

CAD-centric Computation Management System for a Virtual TBM

Phase-I SBIR Final Report
HyPerComp Report Number: VTBM-2008-P1

HyPerComp Inc.
2629 Townsgate Rd., Suite 105
Westlake Village, CA 91361

Topic number 31 (b) Blanket Materials and Systems
Phase-I grant number: DE-FG02-08ER85199

Authors:

Ramakanth Munipalli, K.-Y. Szema, P.Huang, C.M. Rowell, A.Ying, M. Abdou
e-mail: mrk@hypercomp.net

Abstract

HyPerComp Inc. in research collaboration with TEXCEL has set out to build a Virtual Test Blanket Module (VTBM) computational system to address the need in contemporary fusion research for simulating the integrated behavior of the blanket, divertor and plasma facing components in a fusion environment. Physical phenomena to be considered in a VTBM will include fluid flow, heat transfer, mass transfer, neutronics, structural mechanics and electromagnetics. We seek to integrate well established (third-party) simulation software in various disciplines mentioned above. The integrated modeling process will enable user groups to interoperate using a common modeling platform at various stages of the analysis. Since CAD is at the core of the simulation (as opposed to computational meshes which are different for each problem,) VTBM will have a well developed CAD interface, governing CAD model editing, cleanup, parameter extraction, model deformation (based on simulation,) CAD-based data interpolation. In Phase-I, we built the CAD-hub of the proposed VTBM and demonstrated its use in modeling a liquid breeder blanket module with coupled MHD and structural mechanics using HIMAG and ANSYS. A complete graphical user interface of the VTBM was created, which will form the foundation of any future development. Conservative data interpolation via CAD (as opposed to mesh-based transfer), the regeneration of CAD models based upon computed deflections, are among the other highlights of phase-I activity.

Key words

ITER, Nuclear fusion, Test blanket module, Virtual Environment

1. Table of Contents

1. Table of Contents	2
2. List of Acronyms	3
3. Identification and Significance of the Problem / Opportunity	4
3.1 Multiphysical issues in ITER-TBM design	5
3.2 The need for a VTBM and its scope	6
3.3 Project goals, identification of a peer group	8
3.4 Salient features of comparable codes	10
3.5 A long term vision for VTBM development.....	11
4. Degree to which Phase-I has demonstrated technical feasibility	14
4.1 The evolution of a conceptual VTBM design	14
4.2 A draft graphical user interface for the VTBM.....	23
4.3 CAD-based operations	32
4.4 A case study in coupled code execution	37
4.5 CAD/geometry model development, template based approach.....	42
4.6 The exacting needs of mesh generation for TBM.....	44
4.7 A native format for data storage	46
4.8 Format conversion	48
5. Conclusion and Future Plans.....	49
6. References.....	50

2. List of Acronyms

ADF	Advanced Data Format
ANSYS	Commercial Software from ANSYS Inc.
API	Application Programming Interface
CFD	Computational Fluid Dynamics
CGM	Common Geometry Module
CGNS	CFD General Notation System
DCLL	Dual Coolant Lead Lithium
DEMO	DEMONstration Power Plant
EM	Electromagnetics
FCI	Flow Channel Insert
FNST	Fusion Nuclear Science & Technology
FS	Ferritic Steel
GUI	Graphical User Interface
HCCB	Helium Cooled Ceramic Breeder
HIMAG	HyPerComp Incompressible MHD solver for Arbitrary Geometries
IGES	Initial Graphics Exchange Specification
ISE	Intelligent Simulation Environment
ITAPS	Interoperable Technologies for Advanced Petascale Simulations
ITER	International Thermonuclear Experimental Reactor
MCNP	Monte Carlo N-Particle transport code
MDA	Multi-Disciplinary Analysis
MHD	Magnetohydrodynamics
MOAB	Mesh Oriented datABase
MpCCI	Mesh-based Parallel Code Coupling Interface
NURBS	Non-Uniform Rational B-Spline
PbLi	Lead-Lithium alloy
SC/Tetra	Commercial code from Software Cradle Co., Japan
SiC	Silicon Carbide
SIDS	Standard Interface Data Structures
STEP	Standard for the Exchange of Product Data
TBM	Test Blanket Module
TEMPUS-G	HyPerComp proprietary CAD, surface modeling and meshing program
TMAP	Tritium Migration Analysis Program
VTBM	Virtual Test Blanket Module
VTBM-CD	Concept Development version of the VTBM used in Phase-I

3. Identification and significance of the problem / opportunity

There has been a dramatic surge in simulation efforts relevant to fusion technology over the past few years. Commercial software vendors have expanded the breadth of their offerings to include rapid physical modeling, smooth integration with CAD packages and convenient coupling across physical disciplines. Traditional neutronics modeling via MCNP has been extended to arbitrary geometries by providing CAD interfaces. Broadly multiphysical modeling efforts in the area of fusion plasma and structures under radiation have been initiated (e.g., FSP²⁸, VISTA²⁹ among others). These developments will enable a comprehensive study of the fusion reactor environment, thus reducing some of the risk (and thereby cost) associated with experiments. Progress towards the construction of ITER has led to growing international interest in advancing comprehensive simulation capabilities for fusion. The virtual blanket module simulation management software being investigated here bears potential to provide the US fusion program unique capabilities in the research community, and contribute to US leadership in world programs in FNST and ITER-TBM.

The CAD-centric Virtual Test Blanket Module (CVTBM) being investigated in this SBIR project aims to provide an integrated prediction capability for blanket modules operating in a fusion environment. (By the inherent nature of this simulation, the CVTBM could be extended to include other parts of the fusion reactor as well, such as the divertor and other plasma facing components.) The structure immediately surrounding the fusion plasma that forms a plasma chamber (referred to as a blanket,) serves a vital role in fusion energy systems in providing Tritium fuel self-sufficiency, radiation shielding of the vacuum vessel, and efficient power extraction. The physical environment encountered in blanket modules is unique and complex. An extreme combination of physics involving neutron radiation, heat transfer, strong electromagnetic fields, unsteady fluid flow, production and transport of Tritium, structural deformation, render the design, analysis and operation of the blanket module particularly challenging. Work proposed here builds upon prior research at HyPerComp, TEXCEL and the Fusion Science and Technology Center at UCLA leading to an integrated multiphysical modeling environment for blanket physics. Test blanket modules (TBM) investigating various design concepts are to be inserted in the International Thermonuclear Experimental Reactor (ITER) (Fig [1]) right from the start of its projected operation.

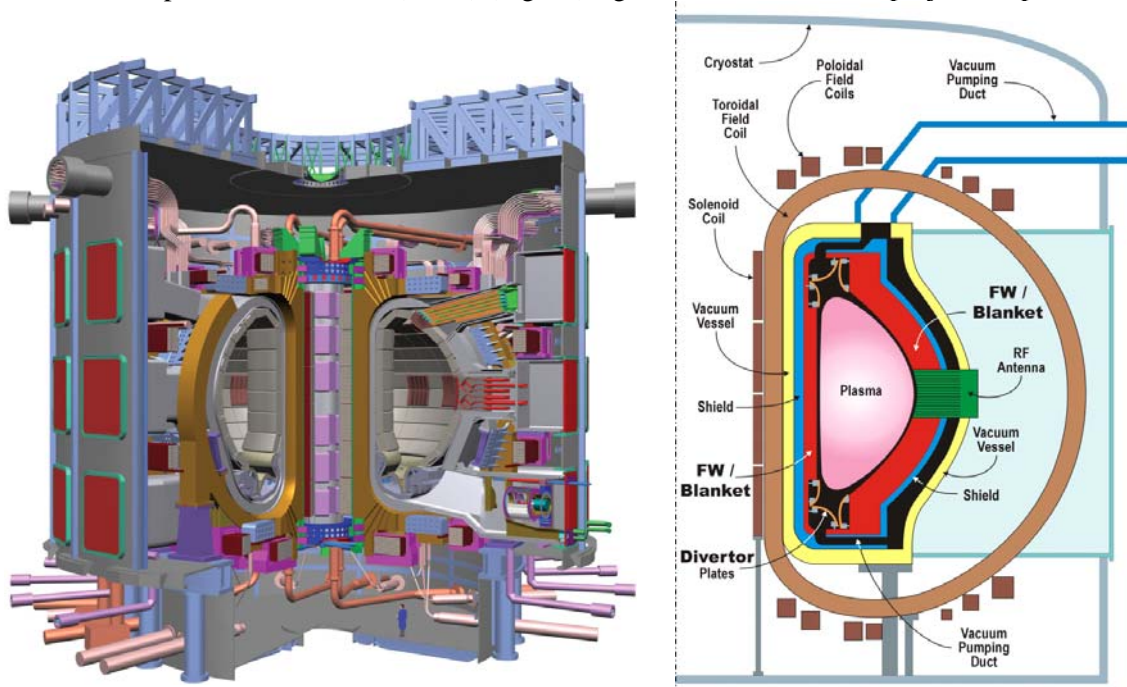


Figure 1: Conceptual sketch of ITER and a cross section showing fusion power technology components

3.1 Multiphysical issues in ITER-TBM design

ITER is an international project that aims to test burning plasma physics in a prototypical fusion reactor environment. One of the objectives of ITER is to test and evaluate modules of blanket concepts for a fusion DEMONSTRATION reactor (DEMO). Test Blanket Modules (TBM) from the various ITER parties will be inserted in three specifically designed ports (see Fig [2]). The primary functions of the blanket are to extract heat at high temperatures and breed tritium at a rate sufficient for tritium self-sufficiency. The primary function of blanket testing is to study the effect of integrated environment on the multi-physical phenomena governing blanket behavior. The blanket includes various materials and interfaces: breeder, coolant, structure, neutron multiplier, and depending on the blanket concept, thermal and MHD insulators. The blanket is exposed to surface heat flux from plasma radiations and neutrons that interact with materials generating volumetric heating and radiation damage effects. Many technical disciplines are involved in analysis and design of the blankets, for example, neutronics, fluid mechanics, electromagnetics, heat transfer, structural mechanics, chemistry, materials science and plasma-material interactions.

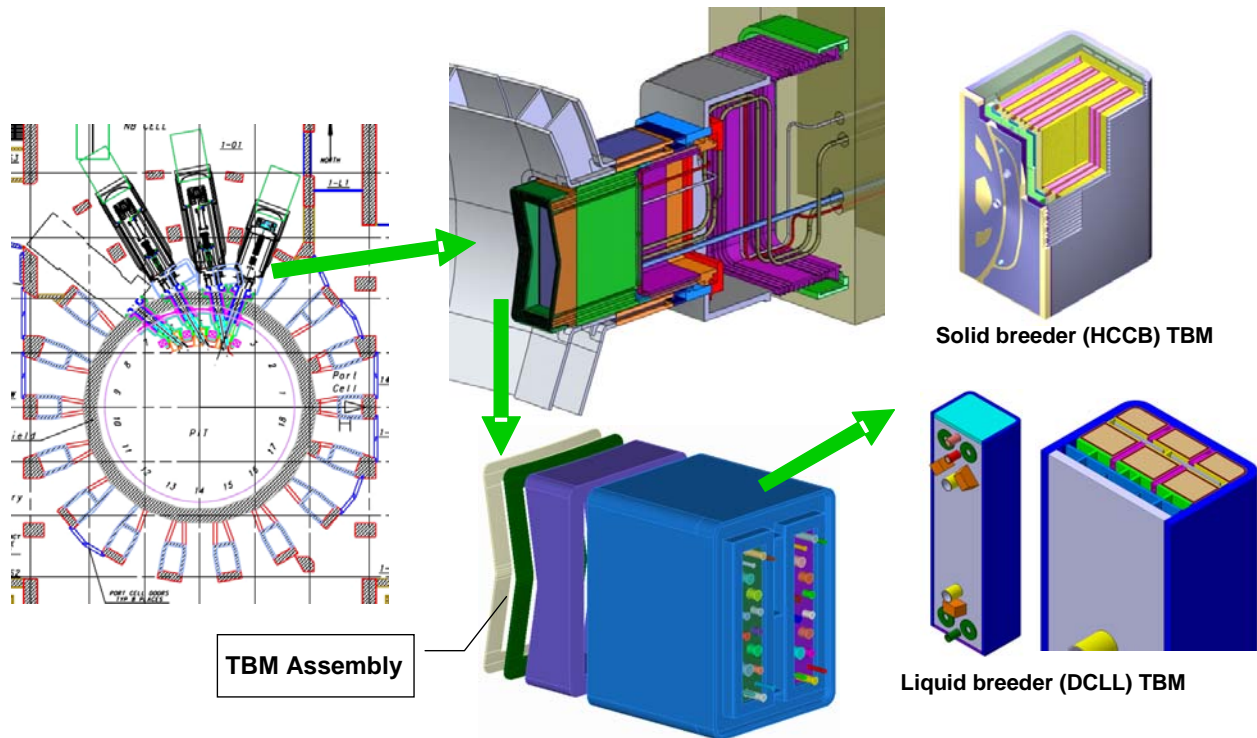


Figure 2: Sectional view of ITER showing equatorial ports where TBMs will be inserted

In this SBIR project, we focus on the Dual-Coolant Lead-Lithium (DCLL) liquid breeder, which has been identified as the US favored liquid breeder blanket concept for testing in ITER (ref. Abdou et al^{1,27}). Here, the first wall and structure of the blanket module are cooled with a separate coolant (e.g., Helium,) while the moving liquid breeder region (e.g., 83% Pb – 17% Li alloy) is self-cooled. Fig [3] shows a sample configuration of the dual coolant liquid breeder concept. This choice of the dual coolant liquid breeder TBM in our study is motivated by two factors. First, it is the primary choice for the US plans for the liquid breeder ITER test blanket module. Secondly, HyPerComp Inc has, under support from DOE developed a high performance computing tool (named HIMAG^{16,30}) that is unique in its ability to model the DCLL concept and can be readily incorporated in the virtual test environment being created. Some work has already been performed in this direction in a previous project at HyPerComp, as discussed in section [8] and ref [30], and can effectively be leveraged to assist the proposed development.

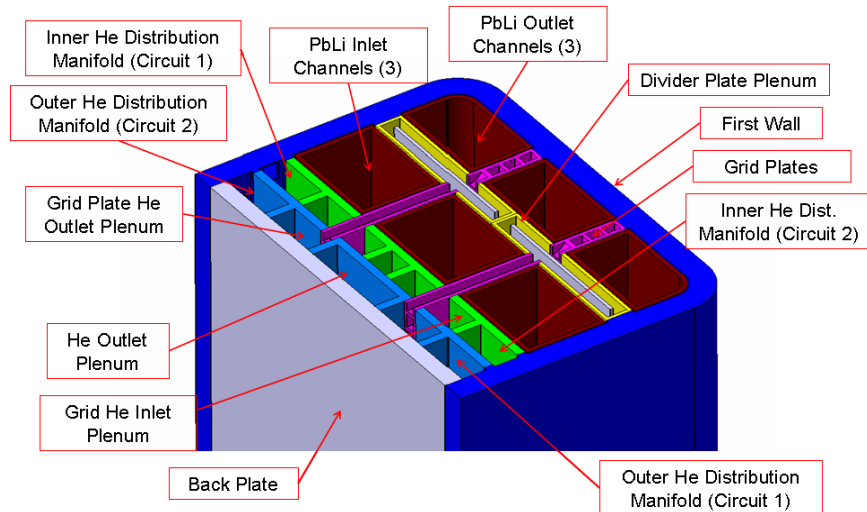


Figure 3: Cross sectional view of a liquid breeder dual-coolant TBM

3.2 The need for a VTBM and its scope

The development of a visual computing environment such as the VTBM is essential for computer based design and testing in virtually every field of engineering. The software described here addresses a contemporary need in the fusion community in light of the ongoing ITER related activities, and can lead directly into product development in collaboration with members from the manufacturing sector. The ability to carry out virtual experiments on computers for a small fraction of the expense and risk involved in the laboratory testing of complex devices is of tremendous importance to all areas of engineering design. It is essential that such procedures be sought in the technology intensive nuclear and aerospace industries. The range of technical innovations leading to effective design of fusion blankets (Ref. Abdou et al¹, Tillack et al²²) mandates a firm synergy across complex physical disciplines, and strongly justifies the need for a virtual computational environment for concept testing. The VTBM as proposed here aims to manage a series of high-fidelity computer programs dedicated to each relevant physical discipline, by providing a convenient engineering interface, and physically consistent data transfer across the various codes.

Fig [4] shows the steps that are typically followed in the thermal-structural analysis of TBM concepts. While there are various other components to the simulation process, this initial example illustrates a few important features:

- (1) Various codes, modules and data file formats are involved at each stage of modeling;
- (2) The work flow is sequential – a coupled solution will require iterating the multiphysical solution until convergence is attained
- (3) A familiarity with third part software to the extent of producing reliable solutions specific to the TBM is assumed – this implies not only the basic learning of software, but also understanding the sensitivities of the numerical procedure related to the specific physics of the TBM
- (4) When there are vast disparities in length scales, (e.g., Hartmann layers in MHD are only microns wide, while structural phenomena in the TBM do not require that degree of resolution), small errors in interpolation across computational meshes used for various phenomena can cause tremendously differing solutions. This is a major concern in modeling MHD flows at high Hartmann number, which are characteristic of the DCLL concept being addressed here. Experience has revealed that accuracy can only be attained by taking due care to conserve mass, momentum, energy and electric currents.

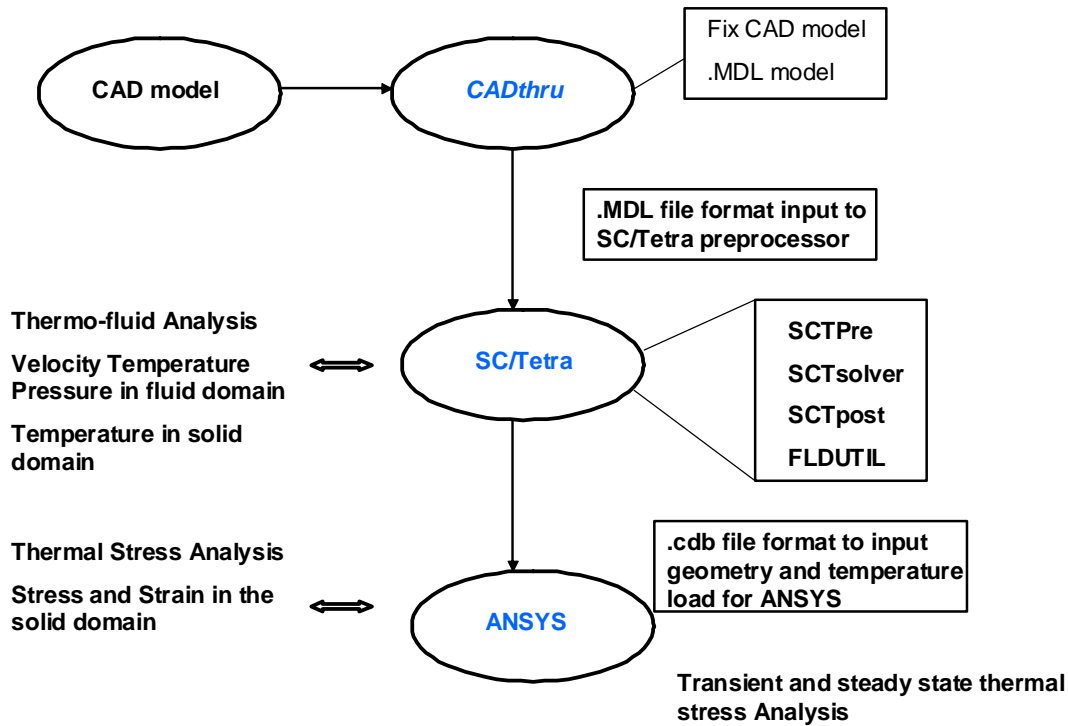


Figure 4: A sample of the work flow involved in thermofluid-structures TBM analysis

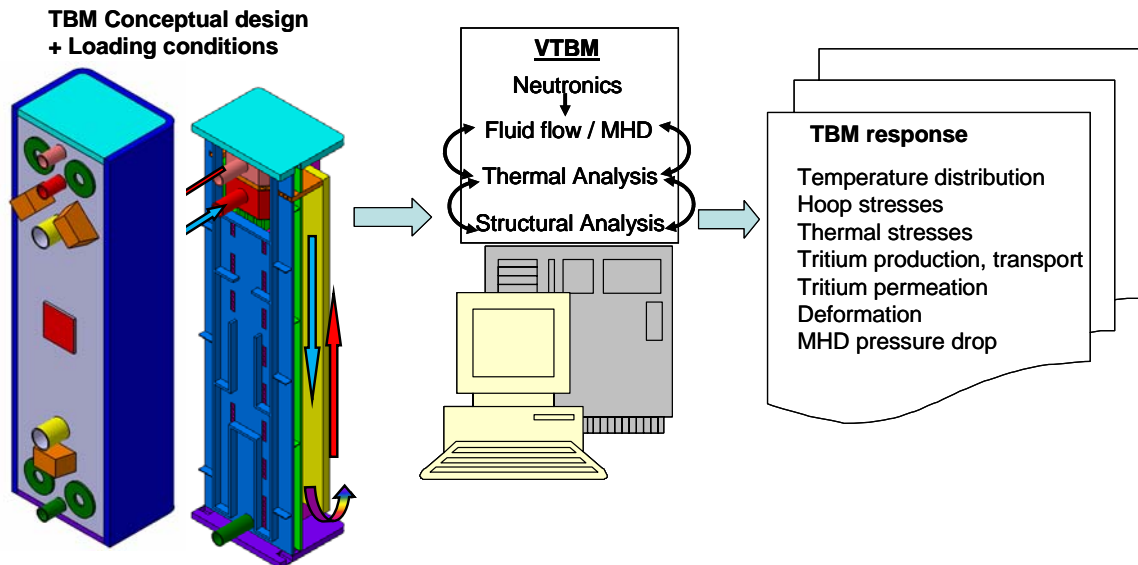


Figure 5: A simplified representation of the function of the Virtual TBM

The actual behavior of a blanket module in the fusion environment is indeed extremely complex. Neutron and other forms of radiation emerging from the plasma core of the reactor are incident upon the first wall – blanket module assembly. The effects of plasma radiation and fusion neutrons include heating of the components; production, transport and permeation of Tritium; and global deformation of the structure. Structural deformation in its turn influences flow and heat transfer, and results in strong coupling of physics. The flowing liquid metal breeder experiences magnetohydrodynamic (MHD) forces that are felt by the structure in the form of hoop stresses. Diffusion and convection processes result in property gradients that influence heat distribution and species (Tritium, corrosion product) concentrations. The

enormity of the MHD (Lorentz) force and its strong dependence on geometric and other design parameters make it the single most important physical effect in the modeling of the liquid breeder DCLL blanket. In this project, we hope to streamline the multiphysical TBM modeling process to the extent that the VTBM facility becomes a test-bed for not only obtaining performance estimates, but to also validate design changes, explain trends, and to trouble shoot errors. An idealized operation of such a VTBM software is illustrated in figure [5]. An integrated environment has been depicted in figure [6], highlighting the key functions of the VTBM software.

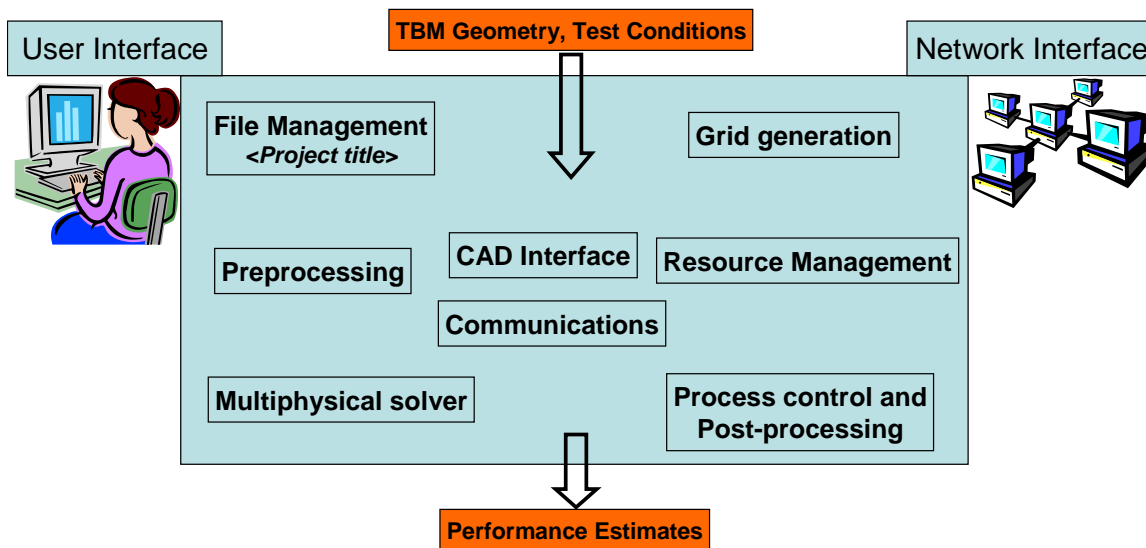


Figure 6 : VTBM functionality

3.3 Project goals, identification of a peer group

The primary intent of this SBIR project is to create a software environment that will manage the multifarious task of simulating the physics of blanket modules in as comprehensive a manner as possible. The software being developed to this end, named VTBM, is envisioned to be a single graphically driven program that can interface seamlessly with a suite of codes that specialize in the modeling of individual physical phenomena, while alleviating certain onerous tasks in problem setup such as managing a CAD-centric simulation process, automatic mesh generation, data import and exchange, effective visualization and troubleshooting, and an accurate computation of blanket performance parameters. A certain amount of “intelligence” would be embedded in the VTBM software, such that the user may avoid common errors in setting up a simulation task and in interpreting its results. Clearly, software such as this can have a very vast scope. The following is a list of features we seek to develop in the VTBM in its role to provide predictive capability for blanket research in ITER and other advanced burning plasma fusion devices:

- Project management – manage the entire simulation process from one convenient interface
- Assistance in problem setup across multiple software platforms
- Time and Space coupling – “loosely” or “tightly” coupled simulation process
- Intelligent problem setup – expert systems, nonsense checks
- Conservative and accurate interpolation techniques – Fast octree approaches
- Variable fidelity modeling: the use of symmetry planes, 1-D to 2-D to 3-D solution projections

The present decade has witnessed a rapidly growing interest in multiphysical simulations and an increased number of offerings from various vendors in catering to this market. While mainstream modeling platforms such as ANSYS, COMSOL (FEMLAB), CFD-ACE+ have demonstrated certain niche

multiphysical modeling capabilities in areas such as aeroelasticity, material processing, and the modeling of small electronic components, a truly multiscale coupled platform that is relevant to the range of physics relevant in fusion blankets which provides adequate flexibility, ease and accuracy is not available in the present time. Further, due to the reluctance from user groups in the industry to adopt a comprehensive multiphysical platform in the production process, the use of such software is currently academic in nature. We explain these aspects in more detail in section [5.1], and describe to what extent the VTBM overcomes these inhibitions. For the present, we recognize that there is a significant need for the multiphysical virtual prototyping environment in nuclear fusion (see e.g., Narula et al [7], Fischer et al [5]). Recent progress in linking CAD-modeling ability to legacy software such as MCNP [12] has demonstrated the enormous potential of CAD-centric modeling in fusion technology systems. The VTBM aims to draw from the success of these efforts and greatly augment them by extending the procedure to comprehensive physical modeling and using an effective project management software that satisfies the needs of the fusion research community.

In addition to the commercial vendors mentioned above, a few innovative software platforms have recently emerged in the area of Multi-disciplinary analysis (MDA), with the intent of providing generic capabilities in integrating third party software in a simulation process. Notable among these are:

AML (Technosoft Inc.): Adaptive Modeling Language. Product, process development cycle integration, and multidisciplinary modeling. Knowledge based engineering (KBE) framework that captures knowledge from the modeled domain and creates parametric models. Object oriented environment to model entire product development cycle. Interfaces for mesh and geometry import/export. Third party software plugins. Knowledge based: Software can “learn”.

ISIGHT (Engineous Software): Rapid integration of commercial and in-house simulation programs. Automates code execution, optimization, design of experiments, quality engineering, and visualization.

ModelCenter (Phoenix Integration): Visual environment for process integration. “Adaptable”. Design, archive and update the design process all in a visual environment. Process Data Management (PDM) tools help store information about process and design data.

MpCCI (Fraunhofer Institute): User specified coupling across third party software for generic modeling processes. Code provides very little assistance with problem setup, and merely runs codes in a combined (batch) mode with inter-solver interpolation routines.

MDOPT (Boeing): CORBA based inter-domain communication facility creates workflow criteria for multiphysical coupling and optimization.

CGM, MOAB: Tim Tautges, Paul Wilson – Applied to neutronics, Sawan et al: Vast library of general purpose open source CAD, mesh tools which may be integrated into user applications involving multiphysics coupling. No support for third party software. Present efforts focus on neutronics, though open to others. Most attractive features: Native CAD format support, generalized data structure.

NURESIM, SALOME: (Nuclear Reactors SIMulation, SALOME-open source set of tools for pre and post-processing), European Platform, CEA (France), U.Pol.de Madrid, Spain, Paul Scherrer Instt.,Switzerland. A research environment with long term goals involving a large number of EU research groups. A large emphasis is placed on developing newer physical models and integrating them under a single head – lukewarm about third party software.

MpCCI: Integrated simulation software, Fraunhofer Instt. A set of graphical utilities to launch multiphysics software (including ANSYS, etc.) No support is provided for problem setup, CAD handling,

mesh generation. Essentially a series of scripts automated through an interface. Inter-process communication is managed by defining very precisely interfaces and data to be exchanged.

3.4 Salient features of comparable codes

We present here operational strategies of a few of the software mentioned above. We conducted as part of phase-I work, a brief survey of the usage of such software in industrial environments and present a critique with our redressals in the following sections.

MpCCI (Mesh-based Parallel Code Coupling Interface):

Multidisciplinary modeling using multiple third-party software is accomplished in MpCCI using (in most cases) open APIs (application programming interfaces) of the external codes to establish an exchange of information between MpCCI and the “untouched” codes. MpCCI is linked to these codes as any regular “user defined function.” As shown in fig. [7a], and MpCCI coupling server manages a scripted simulation process by communicating via a series of APIs with individual programs. The server waits for requests from the coupled codes which run across CPUs using MPI. Key features of this coupling are:

- User selects codes, corresponding input files with model data
- Select element groups in codes where coupling occurs
- For each coupled component specify quantity to transfer: e.g., p, T etc.
- Select coupling algorithm: fixed/adaptive time step or couple on demand
- Set neighborhood search configuration, mesh quality checks, output params
- Launch job on network of computer resources:
- Launch MpCCI server first, and then launch the coupled codes in batch or interactive mode.

Interpolation techniques:

- Point-element relations for standard interpolation
- Element-element relations based on intersection
- Point-point relations for matching grids/nearest neighbor

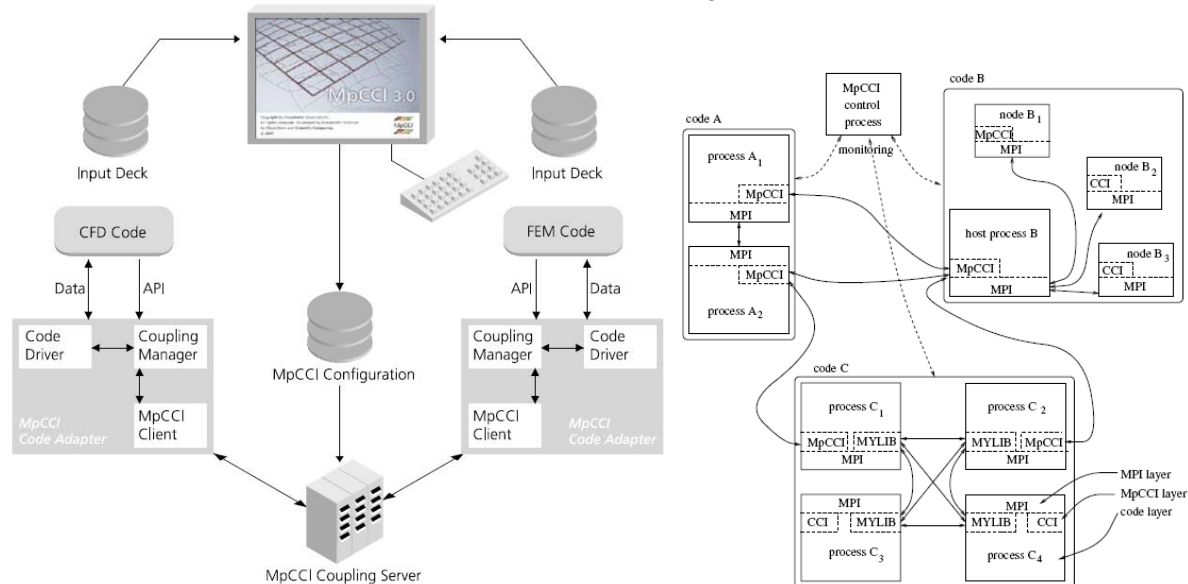


Figure 7: (right) Module architecture of MpCCI and (left) Process view of three coupled parallel codes

MpCCI assumes that the simulation process has been completely defined including specifics for individual solvers, prior to the integrated code execution. It can accommodate a solver’s internal libraries

for the purpose of data exchange, in addition to the traditional MPI based libraries that are generally used. Fig [7b] shows a simulation in which processes not only communicate via MPI libraries but also through a library named here as MYLIB which is internal to a given solver. MpCCI merely provides an external link to other software in this process. A series of adapters are installed along with MpCCI relevant to each external solver. A mapping between MpCCI native formats (for geometry and field data) and third party software is maintained.

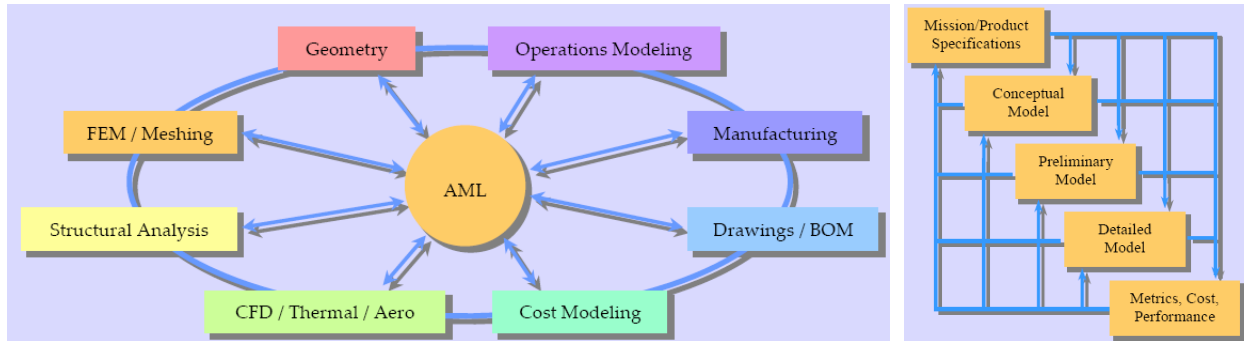


Figure 8: Data management structures from representative MDA software

Adaptive Modeling Language (AML):

A drastically different approach is followed in Technosoft’s AML knowledge based engineering language meant for multidisciplinary and enterprise-wide coupling. AML has an extremely ambitious reach, integrating design, modeling, manufacture, cost and in essence all aspects of production into a single programming paradigm. The Kernel of AML provides language constructs for defining classes, methods and constraint mechanisms. It provides the ability to dynamically instantiate classes and methods and to add, edit and delete objects and properties at runtime. By suitable interfacing, AML has been made to work with CAD based tools and integrated with ANSYS and other external software. It provides an extreme degree of control to the user and considerable effort seems to be needed to customize to a given application, despite the initial intent to automate precisely that aspect of simulation. A schematic of data management structures from MDA software coupled with AML is shown in Fig [8].

3.5 A long term vision for VTBM development

Work proposed here will be performed jointly by HyPerComp Inc. and TEXCEL. The team comprises of leading researchers in computational physics, nuclear fusion and the application of high performance computing techniques to advanced engineering programs. Collaborative support will be provided by ANSYS Inc. and TECPLOT Inc. in supplying software licensing during development, with the agreement that a mutually acceptable commercialization strategy is to be evolved at the appropriate stage.

The development of a VTBM must be based not only on an acute knowledge of the simulation process, but also the needs of a broader engineering community and their prior experiences with such tools. Advanced multiphysical modeling environments such as those noted in the earlier section have suffered from their extreme generality, and an inability to penetrate an industrial environment where a fixed and fine-tuned work flow system is already in practice. Some users indicated a desire to participate in the modeling process in a more hands-on manner than is permitted by conventional coupled physics solvers. During our interviews with some users of such software in the aerospace industry, the following important concerns were raised:

User Concerns

- “Every business needs a great deal of customization”
- “How does one troubleshoot a multiphysical solution: Who is the culprit?”
- “I would like to be able to go under the hood and perform diagnostics. The dash-board type control is insufficient”
- “Existing commercial Multi Disciplinary Analysis (MDA) environments require a lot of customization before they can be integrated into a development cycle”
- “A truly adaptable MDA environment does not currently exist”

There are critical differences in the state of affairs when nuclear fusion research is concerned. Firstly, the physical phenomena encountered in blanket research are characterized by their extreme nature: High magnetic field interaction, strong natural convection currents, large gradients in material properties caused by the presence of radiation, severe material interface effects in heat, current and mass transfer, potential for mild to catastrophic deformations of structure, multiscale coupling ranging from particle modeling codes to continuum flow codes, and structural analysis, (even among continuum processes, length scales vary from sub-micron levels in MHD processes to a few millimeters in structural analysis), and so forth. Much of the science and expertise in individual disciplines rests with research centers, which tend to focus on specific sub-problems relevant to blanket physics. The VTBM could take a leadership role in uniting these various efforts under a single head, potentially operating as a networked-resource for a global user group of specialists.

The VTBM software developed in this SBIR project will have the following distinguishing features.

- 1) Project management: The VTBM will manage the entire simulation project such that no data needs to be manually conducted from solver to solver. Periodic backups and failsafe mechanisms will be enabled. An “under the hood” level of control will be provided for users that seek finer grain control.
- 2) Assistance in problem setup: The VTBM will be equipped with a full featured CAD based solid modeling software which can import CAD data, construct meshes, specify boundary patches and will contain templates to help quickly generate generic geometries that are relevant to the DCLL design. This has the potential to vastly relieve the burden to the user in generating correct and effective meshes. Problem setup will include specifying boundary conditions and input files based customized to DCLL design. Detailed modeling using individual front end programs will be possible for expert users or to aid troubleshooting.
- 3) Heterogeneous, high performance computing: The VTBM will be able to function on heterogeneous computing environments comprising of LINUX, Windows, Solaris, and OS-X. A feature that is being considered for the later stages of VTBM development is one of remote access and execution, thus enabling the VTBM to be a networked resource available to the community at large. A web-based program can handle API calls from remote clients and connect to high performance computational machinery remotely for code execution.
- 4) Visualization techniques: An important contribution of the VTBM could be the visualization of test results, including animation of property contours and structural deformation. Due to the vastly multi-scale and multiphysical nature of data being produced by the VTBM, such a task can be rather elaborate. HyPerComp will be collaborating with TECPLOT (a leading data visualization software maker) in this area.
- 5) Accurate and Conservative interpolation techniques – Fast octree approaches, and NURBS based interpolation techniques relating mesh-based data to CAD models will be used in communicating results across different solvers. This can be critical to the success of coupling physics such as high Hartmann number MHD with structural deformations, due to the extremely large sensitivity of the problem to minor changes in geometry.

- 6) Unsteady Processes: The modeling of the pulsed mode of operation, or that of plasma disruptions, requires a tightly coupled simulation. The code will need an effective eddy current module, and be able to predict deformations in the solid walls caused by eddy currents resulting from the tremendous rate of change of plasma current. This can also result in phase change, cavitation and various other phenomena that require careful analysis. A tight coupling option for time-dependent processes will be inherent in the VTBM.
- 7) Intelligent GUIs, knowledge encapsulation, expert systems: Intelligent agents and expert systems^{14,15} provide the user a friendly and knowledgeable work environment in the simulation process. HyPerComp has developed under NASA support, an expert system enhanced design environment for turbine blades in Boeing – Rocketdyne Propulsion and Power. Such a system reduces potential errors in problem setup and improves the turnaround time in the simulation process.
- 8) Multi-fidelity design and simulation: When the basic framework for the VTBM is in place, we wish to consider the possibility of using models of different degrees of detail to simplify physics and speedup execution. This would enable the code to become a design tool, above and beyond its function as a simulation platform.

During phase-I of this SBIR project, we investigated the basic role and feasibility of a Virtual Test Blanket Module (VTBM) computational system and developed a blueprint for its design. A kernel of CAD-based utilities were developed in phase-I, together with a suite of data translation and interpolation utilities that are compatible with ANSYS, HIMAG and SC/Tetra.

4. Phase-I Goals and Accomplishments

The primary goals of Phase-I research were threefold:

- (a) To evolve a conceptual definition of the VTBM,
- (b) To develop a CAD-centric system for data interpolation, translation and geometry deformation, and,
- (c) To develop a seamless procedure for managing a typical TBM modeling process

These goals have been satisfactorily met, and a framework is now available, which can mature during the course of a potential Phase-II effort into a viable and valuable software utility. We now have a “core” software system around which the high-level functionality of the VTBM may be built. The following are highlights of Phase-I achievements:

- CAD-centric as opposed to mesh-centric modeling
- Third party software (ANSYS, HIMAG, SC/Tetra) gridding, setup, code launching
- Data transfer, interpolation across meshes and CAD
- CAD update, cleanup and deformation
- Automated mesh generation from blocking information
- A unified data format for simulation status
- Detailed GUI design
- Commercialization study – Foresight-ST

The following subsections describe Phase-I work in further detail.

4.1 The evolution of a conceptual VTBM design

VTBM is a simulation management software, whose primary goal is to facilitate the modeling of the multiscale, multiphysical processes in fusion reactors. The present focus is on fusion nuclear components including the blanket: heat transfer, fluid flow, neutronics, Tritium transport, structural mechanics, and electromagnetics/MHD.



We seek to integrate well established (third-party) simulation software in various disciplines mentioned above. The integrated modeling process will enable user groups to interoperate using a common modeling platform at various stages of the analysis. VTBM will strive towards an automated problem setup in a modular manner for all the software it supports. Since CAD is at the core of the simulation (as opposed to computational meshes which are different for each problem,) VTBM will have a well developed CAD interface, governing CAD model editing, cleanup, parameter extraction, model deformation (based on simulation,) CAD-based data interpolation.

Due to the disparate requirements of the various simulation software, data translation and interpolation services will be provided by VTBM. VTBM will provide GUIs to launch and monitor simulations and visualize simulation results. Data pertaining to the entire simulation process will be maintained in a portable format such that user groups may share simulation data and results.

The VTBM will serve multiple functions during the course of a complete simulation. We present in some detail, the manner in which the CAD-core, multiphysical problem setup, execution and post-processing are performed in the following self-explanatory diagrams. Briefly, the CAD-engine received CAD data in a standardized form from an external source, or from a template-based parametric surface model generator, as will be discussed later. If the geometry deforms as a result of the simulation, the CAD model is regenerated. If interpolation of data at surfaces is desired, it is performed here. While at present we are working with neutral CAD formats such as IGES, we wish to transition to “native” CAD formats such as

CATIA and SOLIDWORKS in phase-II, due to the advantages to be gained by a detailed parametric representation of the solid model.

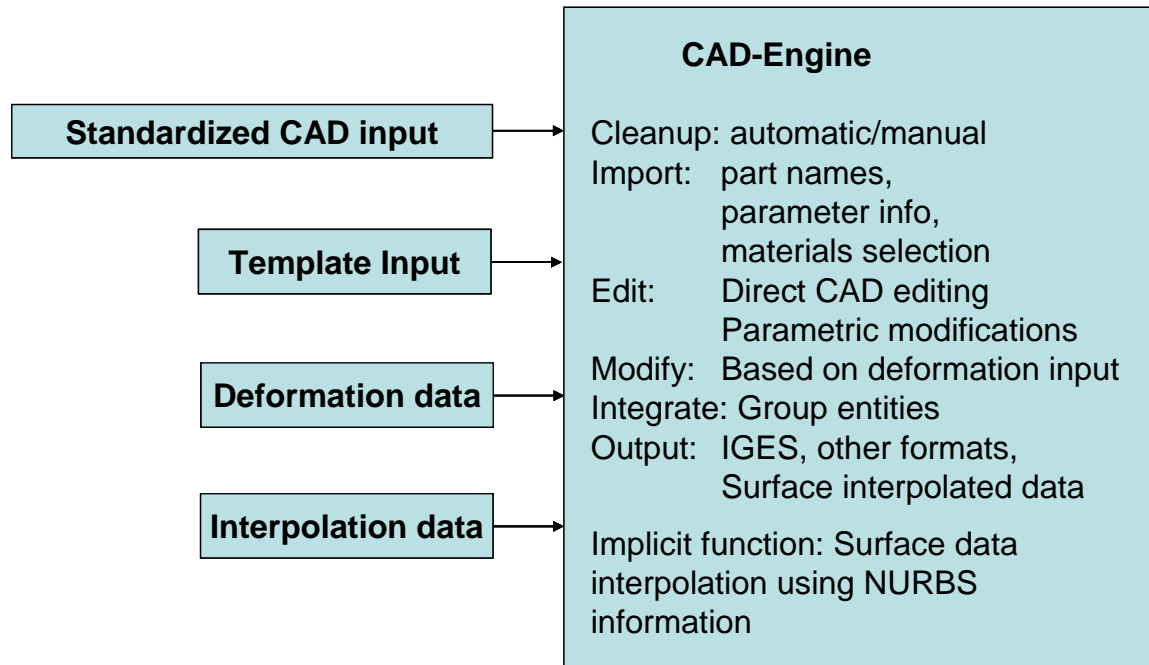


Figure 10: Functionality of the CAD-engine in VTBM

CAD provides the essential geometric information needed for the multiphysical solvers. VTBM uses a CAD engine which can read CAD data, edit it, clean it up, and create a surface or volume mesh depending on physical requirements. In the figure below, we show the flow of grid information (G) from the CAD-grid engine to the various solvers, which in turn produce data such as radiation loading (R), thermal and hydraulic loads, which are exchanged with the other solvers.

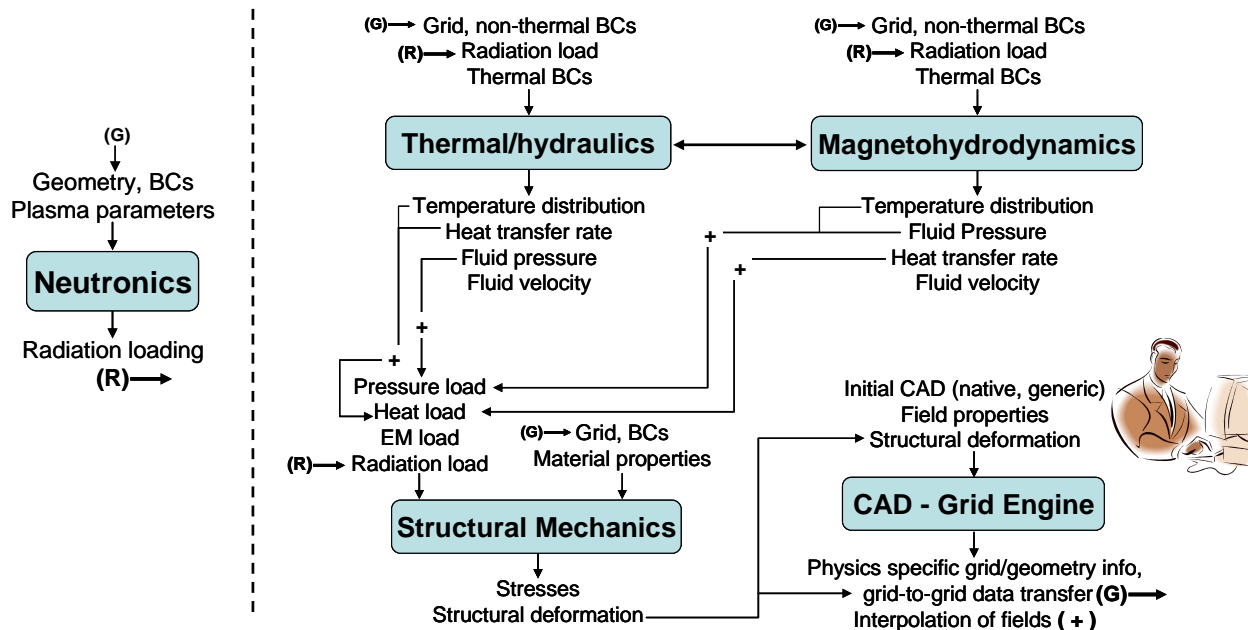


Figure 11: Workflow diagram of the multiphysical solvers in VTBM

VTBM uses a CAD/mesh tool known as TEMPUS-G, which has been developed at HyPerComp (and Rockwell International) over the past 15 years. TEMPUS-G is a very powerful meshing tool, and can produce solver-specific meshes from high-level and friendly user interfaces. This is the first step in setting up solvers. As shown in the following charts, the geometry/mesh utility in VTBM will define line/surface/volume regions in the geometric domain specified in the given CAD model for each physics module (solver). At the same time, regions where different physics modeling codes interact are defined, e.g., external boundaries of a structure, where pressure and temperature need to be provided by the flow solver in order to compute stresses. Mesh resolution at each face is fixed from simulation parameters, e.g., y^+ in turbulent flows, laminate thickness in composite materials, and so forth. Each volume region is meshed, and initial and boundary conditions for the simulation are setup. Input files to control code execution are created, and a data transfer protocol (quantities, units, etc.) at each relevant surface is constructed.

VTBM will execute each solver that has been selected with its appropriate input data, using a coupling strategy that is determined a priori. Code execution will also involve launching the solvers on remote machines, using command line options/scripting techniques to avoid launching the software from graphical interfaces, as well as being able to obtain diagnostic and post-processing information from those solvers using an adequate knowledge of the scripting language of each solver. This requires that each new module to be integrated in VTBM will need to go through a period where plugins are created that will enable the module to interoperate with the rest of the environment. Alternately, an API is created, using which third party vendors may create VTBM interfaces.

Solver Setup	Execution
<p>For each solver:</p> <p>Geo/Mesh</p> <ul style="list-style-type: none"> Import mesh Copy prior setup Define/confirm region IDs Define regions for POST Define contact regions Define resolution needs Mesh each volume <p>Pre-processing</p> <ul style="list-style-type: none"> Setup BCs on each surface Setup initial conditions Setup input file <p>Data transfer</p> <ul style="list-style-type: none"> Interpolation settings (at each interface) 	<ul style="list-style-type: none"> Define compute nodes Verify licenses, access Define execution order Interpolate data in contact regions (as needed) Run first solver with input file If there is no deformation: <ul style="list-style-type: none"> Proceed with INTERPLT Write POST data If deformation exists <ul style="list-style-type: none"> Call CAD, redo geometry Regenerate all meshes Proceed with INTERPLT Write POST data Write debug data Transmit interpolation data Check for user intervention Run further solvers sequentially

Figure 12: Functionality of the solver and execution kernels

Post Processing (POST)

- Check POST region list
- Identify solvers with surfaces
- Verify availability of data
- Check availability of convergence data from each solver
- Following choices made on UI:
 - Display 1-D, 2-D, 3-D plots,
 - Time histories,
 - Convergence histories,
 - Code execution status,
 - Additional debug information
- Export visual data to file(s).

Figure 13: Functionality of the solver, execution and post-processing kernels

Post-processing and visualization is an important function of the VTBM. As shown above, this involves extraction of the appropriate data, preparing time histories and computing key performance parameters, while being able to export graphical information. There is an inherent sequential logic in the TBM modeling process. We begin by describing the conventional simulation as practiced in the present time.

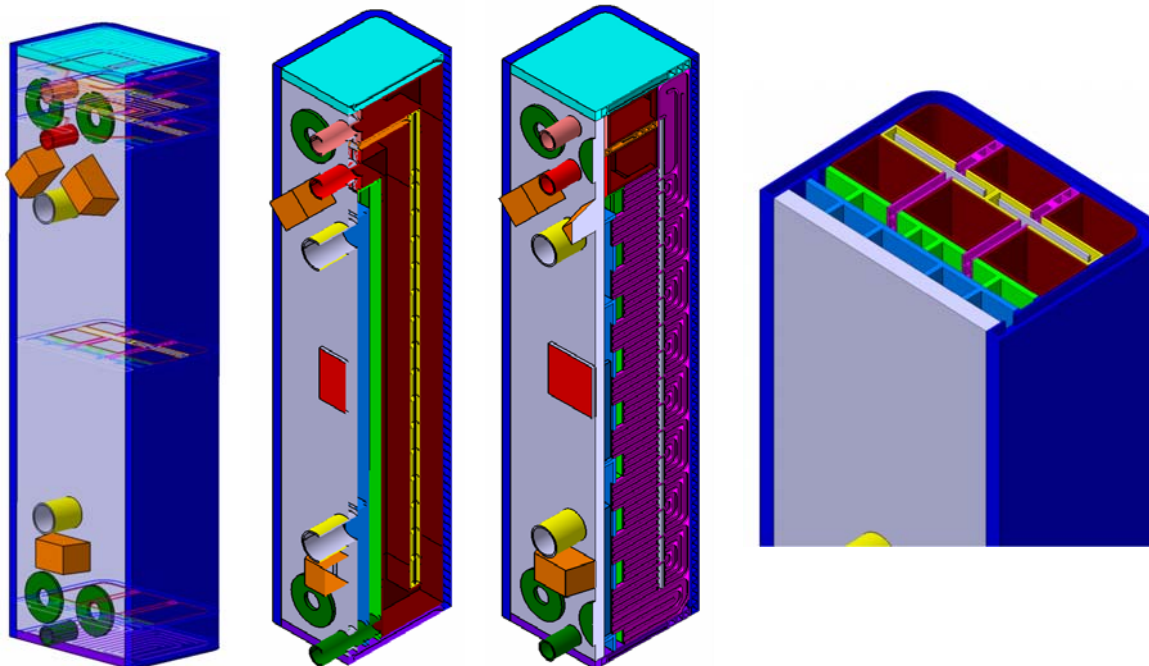


Figure 14: CAD Model for DCLL showing cross sectional views, from Dagher et al [4]

CAD Model Development

The CAD systems CATIA, Pro-E and SolidWorks have been used extensively in developing geometry information for TBM research by various groups. Neutral formats such as STEP and IGES are exported from these programs and are read by the field solvers, and after ensuring that the model is watertight and

that there are no gaps or other defects in the CAD, are then used to generate a computational mesh. Some programs have a direct link to CAD software (e.g., CFDesign provides a button in the CATIA user interface to directly export the relevant geometry to the CFD solver). Indeed, the complete CAD model of the DCLL assembly tends to be complex, as shown in Figure [14]. Some discretion and a uniform naming convention need to be enforced at this stage such that multiple disciplines which require this information may access it with relative ease and inter-operability.

Neutronics

A linear Boltzmann transport equation (LBTE) is solved to model neutron and gamma radiation. This equation expresses the conservation in a differential control volume and accounts for the streaming, collisions, scattering and fixed sources of particles. There are two principal approaches followed in obtaining solutions to this equation:

Deterministic Approach: The conservation law is expressed in a discrete form over a computational mesh, a set of energy levels and sweep angles. Conventional high order accurate finite difference and finite element methods are used in solving the equation set. This method has the ability to provide neutron transport field data at each mesh point in the computational domain, while suffers from various approximations that are used and its enormous memory requirement. Existing codes in this area are:

DANTSYS, PARTISN and DOORS: 1D, 2D and 3D Finite Difference Solvers

ATTILA: 3-D Finite Element Solver with CAD coupling, being presently evaluated for use in ITER.

Statistical Monte Carlo Approach: The Boltzmann equation is solved using statistical methods. Particle travel distance and interaction physics data is converted into cumulative distributions, which are sampled appropriately to obtain transport data at a given location. There are no approximations used in such models for both the sources as well as the transport process. However, this method is computationally very expensive. Variance reduction techniques are needed for better accuracy, the code must be run multiple times to obtain field data in a given region and large CPU time needed. Popular codes based on this approach are:

MCNPX: A Monte Carlo code that originated in Los Alamos and is commonly used worldwide,

MORSE, TRIPOLI and TART

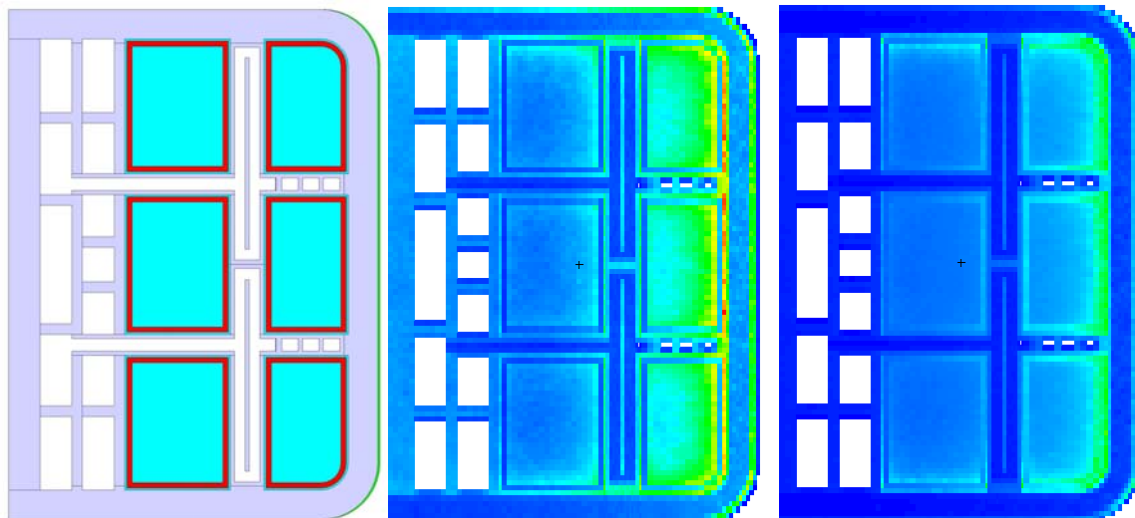


Figure 15: (a) DCLL-TBM Cross section, (b) Computed Gamma heating, (c) Neutron heating contours (ref. Sawan et al [20])

Much progress has been made in recent times (see e.g., Direct Accelerated Geometry – MCNP of Sawan et al [20]) in adapting MCNP to complex geometry situations. Traditional versions of the code were

designed to handle canonical geometrical shapes for enclosures (spheres, cubes, etc.) and some combinations of them, while the coupled CAD procedures have extended the scope of these calculations by enabling an automatic CAD to MCNP facility which is able to decompose a complex geometry into data that is usable in MCNP. Sample results are shown in Fig. [15] for a full DCLL geometry application from ref [20]. While the neutronics modeling is in itself a vast and complex undertaking, certain preliminary estimates are readily provided by neutronics computations. These include one-dimensional variations of power dissipation caused by neutron transport, as shown in fig. [16a]. Such estimates are used in performing fluid, thermal and structural mechanics calculations. In the event that more detailed neutronics data is available, it will be possible to integrate it into the VTBM using the structure developed for the other physical modules. We have started our modeling efforts using the one-dimensional estimates.

CFD/Heat Transfer:

CFD has an important role in modeling heat transfer, the flow path of the liquid coolants (typically Helium and water), as well as in the study of MHD (see below). The complete CAD model of the TBM, including the solid and the fluid parts, is input to a CFD analysis package, which provides the transient and steady state temperature field in the solid and fluid parts as well as the complete 3D flow distribution of the coolant in the manifolds and the coolant channels. Candidate CFD solvers in popular use among the community are:

CRADLE SC/Tetra from Cradle Software Co., Japan, and
CFX from ANSYS

In phase-1 we developed format converters, and input file generators for both of these software, such that we are able to interface with them externally using their native APIs. In general, a coupled CFD, Heat Transfer and MHD solution may be desirable due to the strongly inter-dependent nature of these phenomena. A limiting factor may be that the computational time needed for a converged MHD solution may be in general far greater than that for the thermal and non-conducting fluid flow regimes.

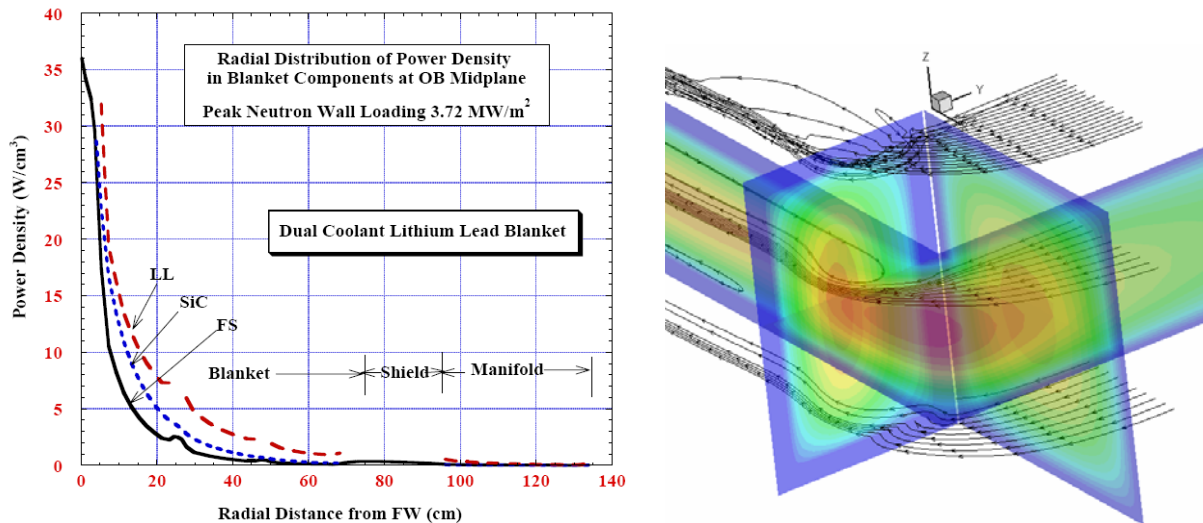


Figure 16: (a, left) Power density given as a function of distance, and (b, right) Temperature contours shown alongside streamlines from an MHD calculation of the DCLL

Magnetohydrodynamics (MHD)

Neutronics codes (conventionally, MCNP described above) provide a guideline of the variation of neutron power dissipation as a function of distance from the first wall. An illustrative result is shown in Fig. 16. This serves as a source term for the MHD part of the solution. HyPerComp has developed a unique software named HIMAG (described more fully in section [8]) written primarily for flows involving high magnetic field interaction, as encountered in the DCLL concept. HIMAG is able to address issues in heat

transfer, strong natural convection, variable properties, arbitrarily conducting walls with contact resistance, anisotropic conduction and ferromagnetic effects at the present time, all of which having a direct bearing on this project. It must be mentioned that conventional approaches towards the numerical modeling of liquid metal flows in DCLL suffer from a loss of accuracy or generality when dealing with complex geometry. In this regard, HIMAG is a sole contender for the MHD part of the solution in VTBM. Efforts are underway to equip HIMAG with the infrastructure to operate in this role. Fig [16b] shows a sample result from HIMAG with temperature contours superposed over streamlines of MHD flow entering a flow channel insert. Fig. [17a] shows MHD flow entering a SiC flow channel insert in a DCLL concept where the insert is encased in a Ferritic Steel outer channel. Three dimensional entry effects in the flow of current and the corresponding streamline pattern are shown in Fig. [17b].

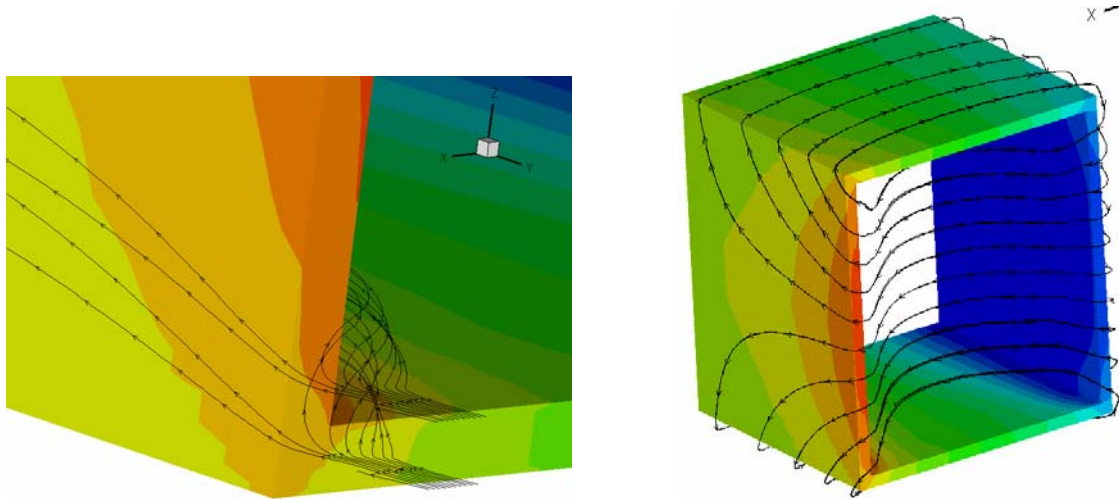


Figure 17: (a) Streamlines and (b) electric current lines are crucial in understanding MHD flows

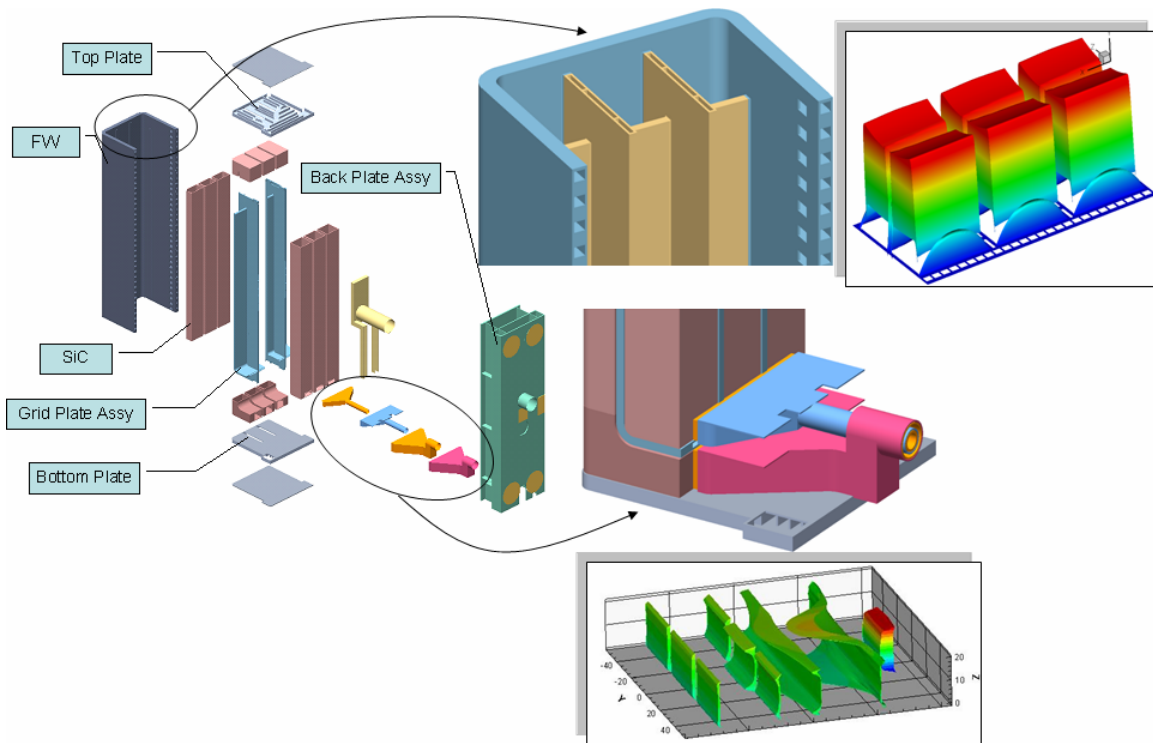


Figure 18: DCLL Blanket details and MHD flow features

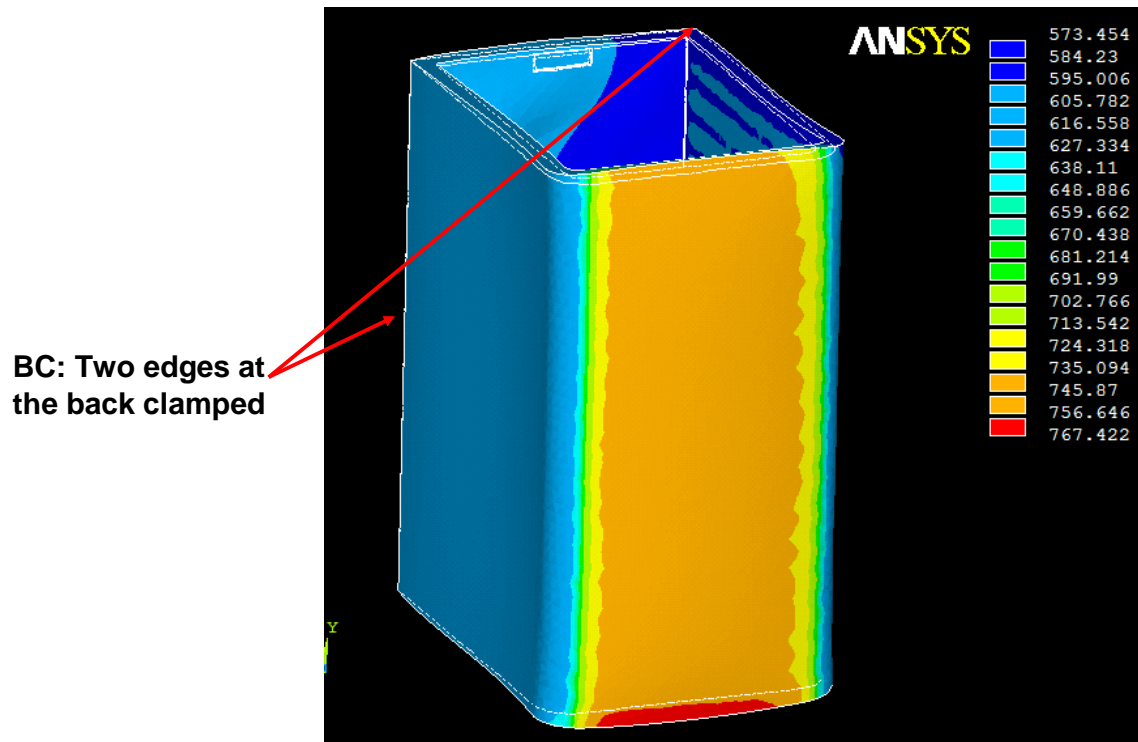


Figure 19: Structural modeling of the first wall manifold using ANSYS showing boundary conditions

Structural Thermo-Mechanics

Thermal and hoop stresses and deflections of the structure result from the intense MHD related pressure loads, as well as heat loads from the strongly radiative environment. While conventional structural mechanics solvers such as ANSYS are able to handle structural deformation, it is likely that a fuller treatment involving the effect of radiation may be needed in TBM modeling. Sharafat et al²⁹ present the initial development of a Virtual International Structural Test assembly (VISTA) concept to perform virtual experiments in the effects of damage as a function of time in the response of complex structures exposed to high intensity radiation. While we do not anticipate radiation damage to be the primary focus of the VTBM study in the initial stages, this research could complement the proposed effort in addressing a broader class of problems in nuclear engineering. Figure [19] shows a typical simulation performed using ANSYS in modeling structural mechanics of the first wall.

Tritium Transport

Tritium permeation is a key safety issue that is receiving increased attention in fusion and in the ITER TBM. A key element in this problem is mass transfer, which is closely coupled to MHD fluid flow and heat transfer. Tritium is implanted on the first wall with an approximately uniform flux across the surface. Hydrogen isotopes are produced in the bulk of the first wall and blanket due to the (n,p), (n,d), (n,t) and other reactions caused by neutron irradiation. Neutron irradiation also produces vacancies and defects in the wall and blanket materials which act as traps for the hydrogen isotopes that are diffusing. The transport of Hydrogen isotopes is dominantly in the direction normal to the incoming radiative flux (and the first wall surface). TMAP (Tritium Migration Analysis Program) is the most widely used software to compute Tritium transport. A preliminary capability to compute Tritium transport has also been provided in HIMAG.

Electromagnetics

During disruptions and other unsteady plasma modes, high strength, highly coupled reaction of conducting media to time varying plasma properties can include melting and other phase change

phenomena. Specialized codes such as ANSYS, FEMLAB and others may be able to handle this, and the time-dependent treatment of electromagnetics in HIMAG can also be used to model some of these phenomena in conjunction with an appropriate structural mechanics solver.

Other Physics

A more comprehensive list including the effects of radioactivity, decay heat, chemistry, neutron-material interaction, thermal hydraulics, thermo-mechanics, thermodynamics and reliability studies among others may be considered for advanced versions of the VTBM software.

Code operation, post-processing and performance estimates

The TBM modeling process will receive as its input, geometrical details about the VTBM specified to a rather fine level of detail (as can be seen in fig. [18]). The CAD models thus obtained are tagged for appropriate component-wise breakdown, and any defects and imperfections are cleaned. Boundary and initial conditions are specified and a flow-heat transfer simulation is performed. This requires an input from the MHD solver giving the velocity profile and heat profile within the liquid metal region, the rates of heat transfer from all boundaries and the rates of heat deposition within the coolant and other solid media. Temperature and pressure distribution thus obtained are fed into the structural analysis solver, which then provides stresses and deflection data for the structure. If needed, the geometry is deformed, re-meshed and a new iteration is performed until the solution is seen to converge to a steady state value. It is likely that there may some times be truly unsteady solutions wherein the coupling process becomes far more sensitive to the way in which data is transferred across disciplines.

Physics	Analysis code	Computational Scheme	Mesh specification	Geometry to Analysis preparation
Neutronics	MCNP	Monte Carlo Simulation	Particle in Cell (PIC)	CAD models have to be imprinted and merged before import into MCNP. Need to incorporate large scale 3D fusion device model (e.g. 40 degree toroidal sector). Direct geometry method does not require CAD translators for MCNP but requires modification of MCNP code
	Attila	Finite Element	Unstructured tetrahedral mesh (node based)	Standard CAD data import. Need to incorporate large scale 3D fusion device model (e.g. 40 degree toroidal sector)
Electro-magnetics	OPERA	Finite Element	Unstructured Hex/Tet mesh (node based)	Requires external meshing tool (CUBIT). Need to incorporate large scale 3D fusion device model (e.g. 20 degree toroidal sector)
	ANSYS	Finite Element	Unstructured Hex/Tet mesh (node based and edge based formulations)	Need to incorporate large scale 3D fusion device models (e.g. 20 degree toroidal sector)
Thermo-Fluids	SC/Tetra	Finite Volume	Unstructured hybrid mesh (node based)	CAD model to mesh requires intermediate translators. The translators modify, clean and convert the geometry into the required input format for the specific analysis code. Special mesh considerations needed to capture boundary layers (correct application of turbulence models)
	CFdesign	Finite Element	Unstructured hybrid mesh (node based)	
	FLUENT	Finite Volume	Unstructured hybrid mesh (cell based)	
MHD	HIMAG	Finite Volume	Unstructured hybrid mesh (cell based)	Requires external meshing tools. Special mesh requirement to capture fine MHD boundary/shear layers
Structural Analysis	ANSYS	Finite Element	Unstructured second order Hex/Tet (node based)	Requires second order elements for accuracy. The elastic stress and strain analysis is followed by a plastic analysis where needed. Fusion relevant material properties and constitutive equations have to be incorporated into the analysis
	ABAQUS	Finite Element	Unstructured second order Hex/Tet (node based)	

Table 1: Details of computational analysis tools to be managed by the proposed VTBM

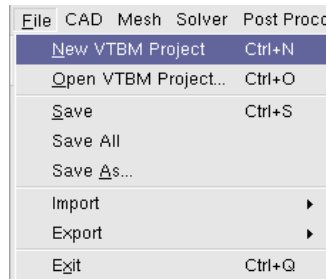
Table [1] shows a sample list of codes that are frequently used in blanket/heat shield analysis. Among these, HIMAG has been developed at HyPerComp Inc. specifically for the purpose of solving MHD flows with strong interaction and high Hartmann numbers relevant to blanket modules. The others are third-party software, and appropriate licensing arrangements must be made while interfacing them into the VTBM being developed here. Negotiations with ANSYS and TECPLOT (for data visualization) have commenced, and it is hoped that a final product will contain bundled versions of a rudimentary set of commercial codes for principal functions. Both of these organizations have expressed an interest in being part of such a venture.

4.2 A Draft Graphical User Interface for the VTBM

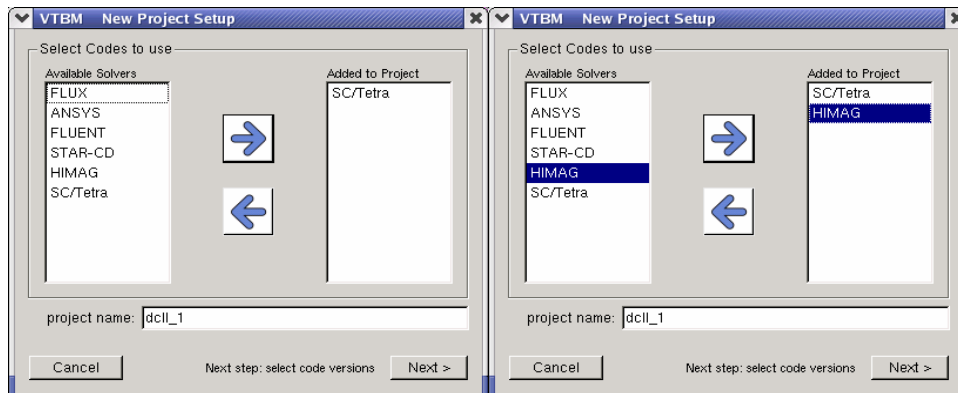
We created a draft user interface for the VTBM which we hope to redefine and refine at the onset of phase-II investigations based upon a user survey. In this section we present highlights of this interface, beginning with the following view of the top-level graphical user interface (vertically compressed!) of the VTBM. We note that the functionalities such as “File”, “CAD” and so forth are in close agreement with the kernels described in section [5.1], and the GUI is not native to any single solver, but rather a full featured geometry manipulation center.



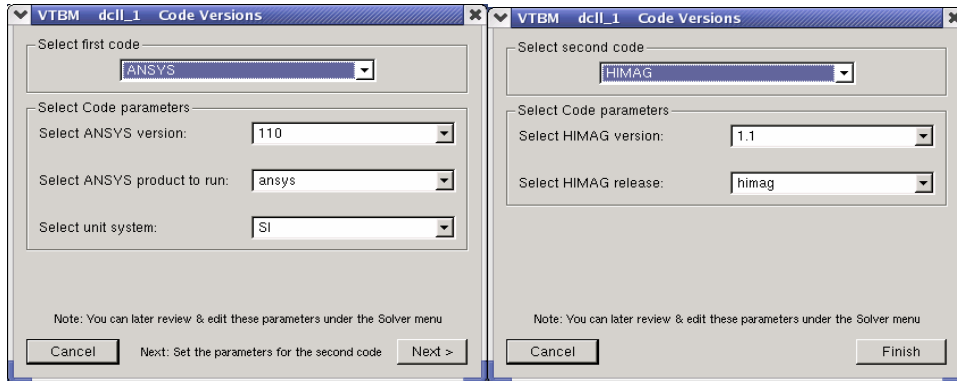
File menu -> New project



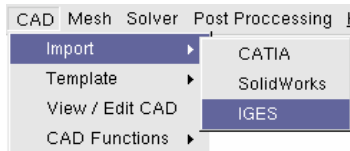
The initial task upon creating a new VTBM project using the “File” menu would be to name the project and select physics modules that need to be modeled, by choosing from a list of available choices (populated at the time of installation or update).



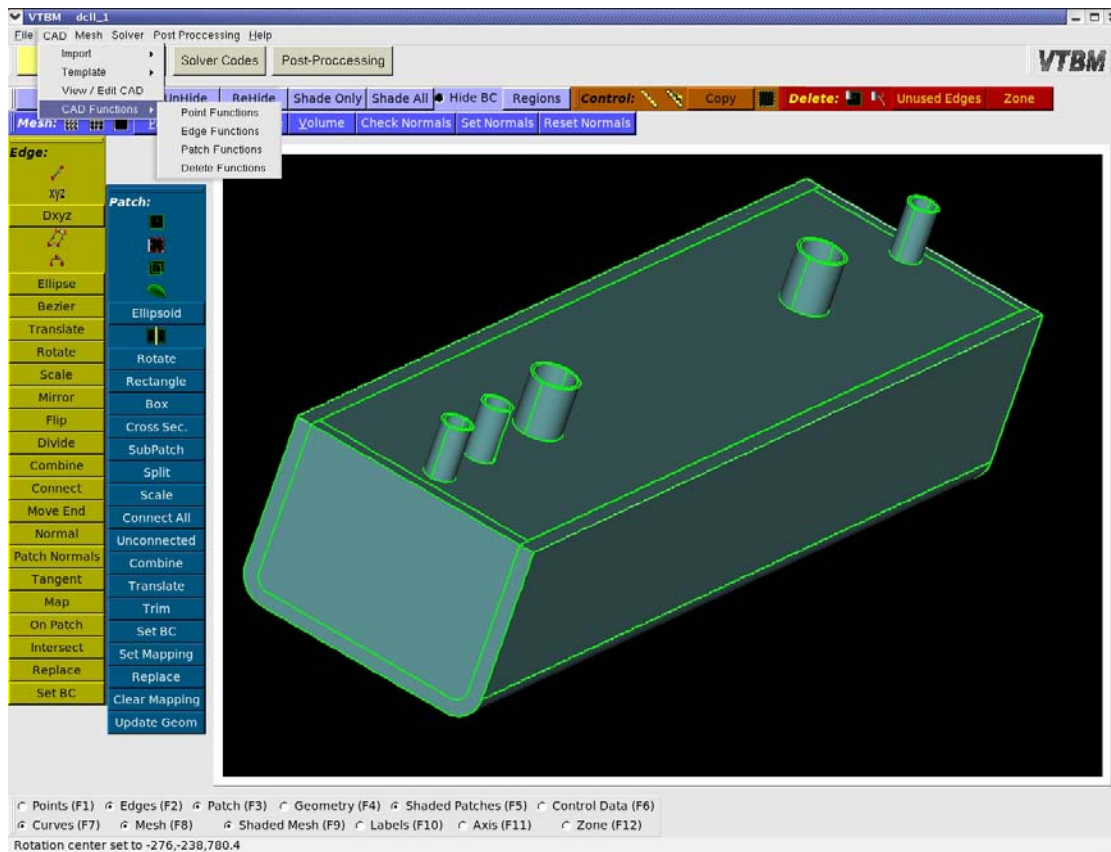
Code modules and their versions are selected at this stage using the following dialog box, which will also prompt for locations of license files and computers and establishes the ability to use these resources. If not, it will simply prepare problem setup for the modules that are inaccessible.



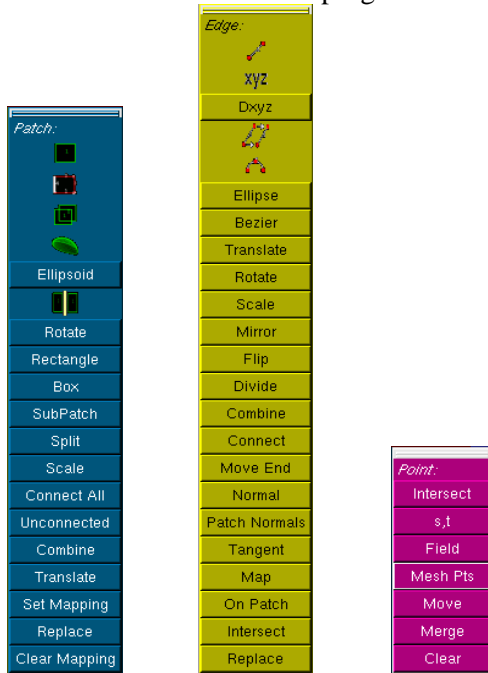
The CAD menu will attempt to load a CAD file into the current project using a set of allowable formats, as follows:



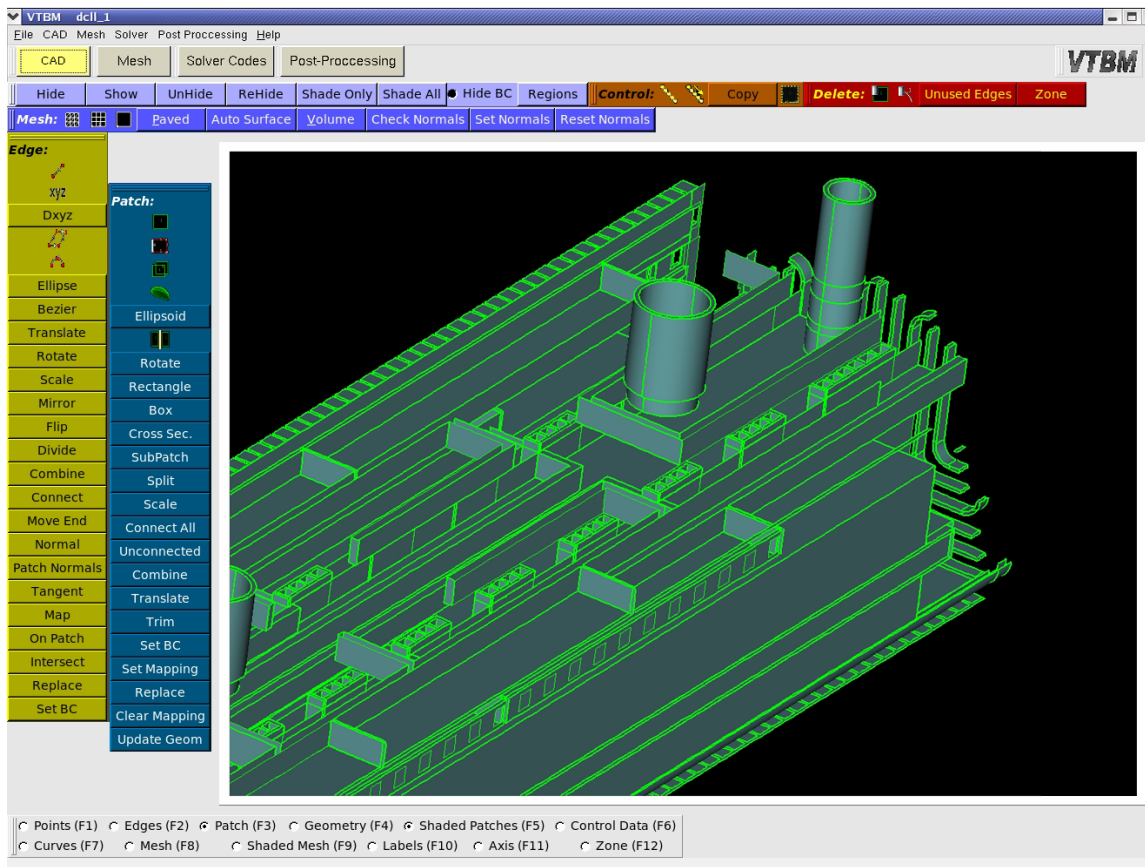
Once the file is imported, it looks like the following screen capture. Note the CAD menu is open showing the organization of the CAD Functions (toolbars).



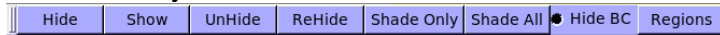
Edit-toolbars for the CAD program are shown in some detail here.



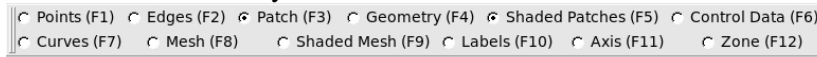
After some patches are made to hide the model is easier to view as shown below: Note the visibility toolbar at the bottom left of the screen has just Patches F3, and Shaded Patches F5 selected.



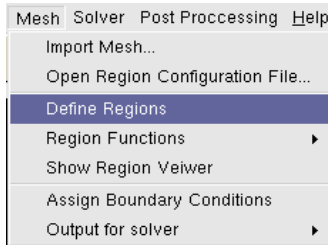
CAD Visibility Controls the toolbar from above screen:



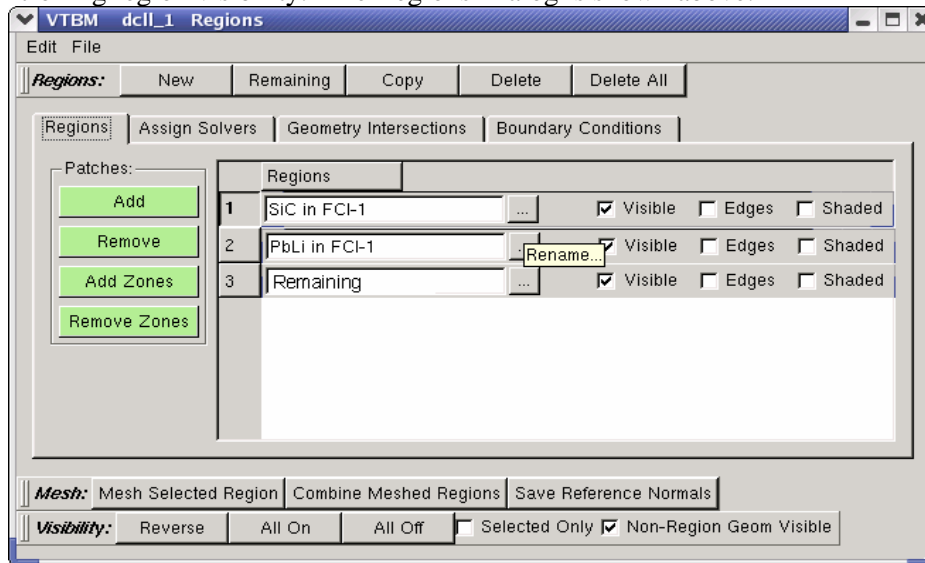
A detail of the visibility toolbar is shown below:



Using the model defined in section [5.1] we examine the “Mesh” dialog box here.



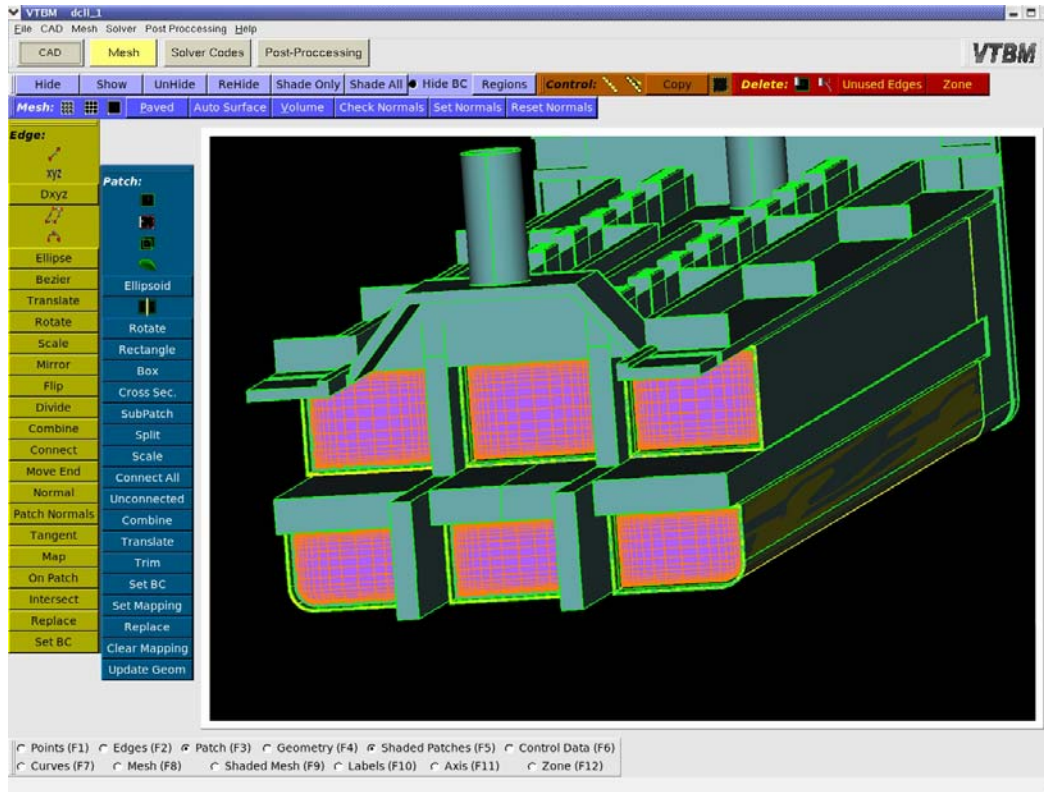
To start we will demonstrate how we create regions and facilitate the viewing/handling of complex models by controlling region visibility. The Regions Dialog is shown above.



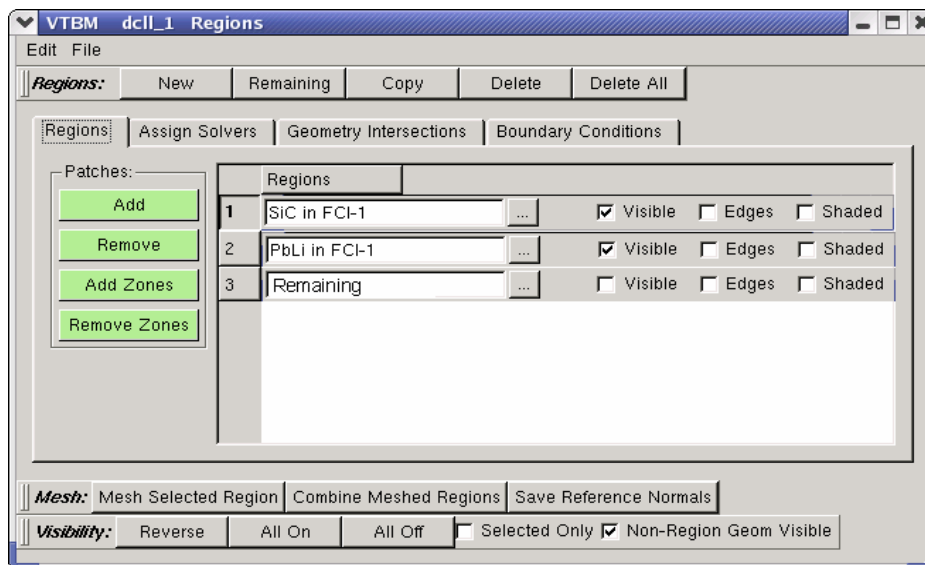
In this screen capture, three regions have been defined. The first two will be used to create a coupling region later, the later region is used to group the remaining elements of the model for turning off visibility as we will see later. Note the *Regions* toolbar with the buttons “New” “Copy” and “Delete” among others. This is where new regions are created.

Note too the Patches Group box: this where the selected Region could add/remove patches from its definition.

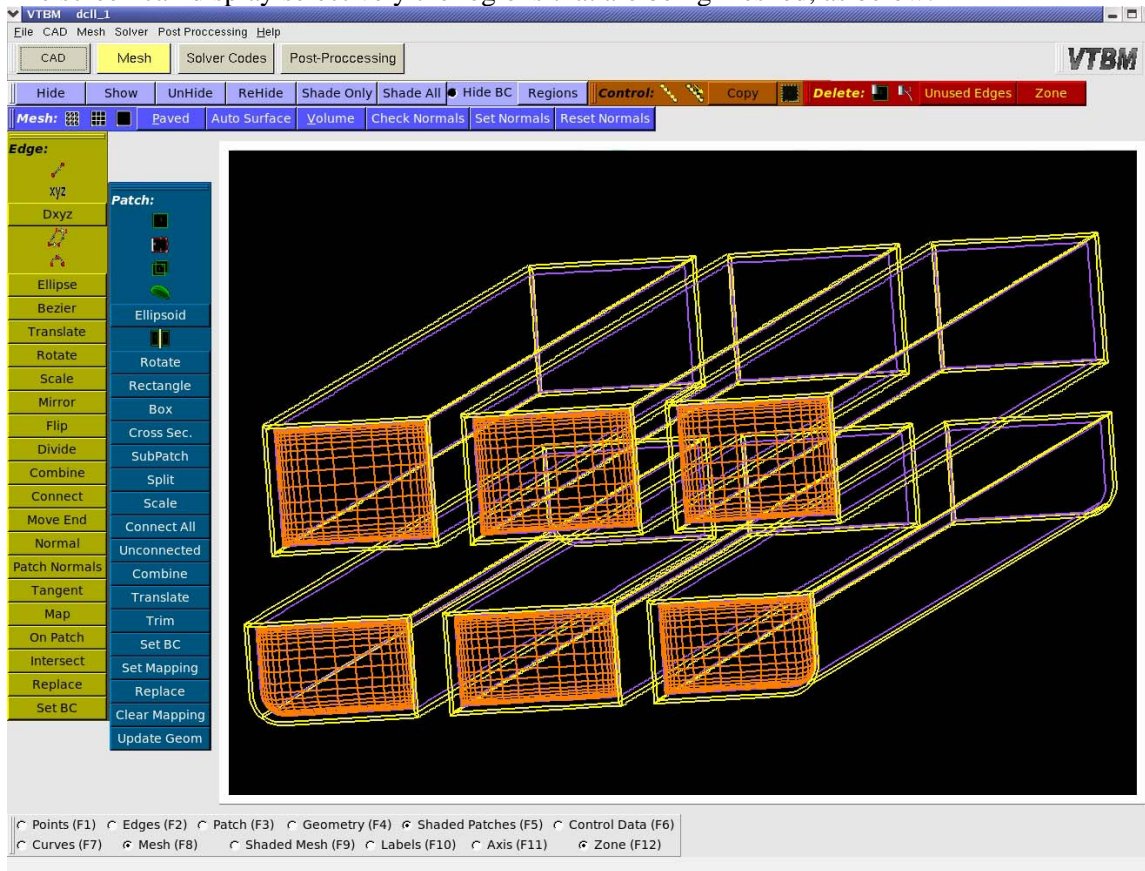
The following screen capture shows details of the three regions defined above. These have been selected graphically, and meshing parameters are assumed to have been specified.



Visibility of each region can be controlled manually, as in the following dialog box:

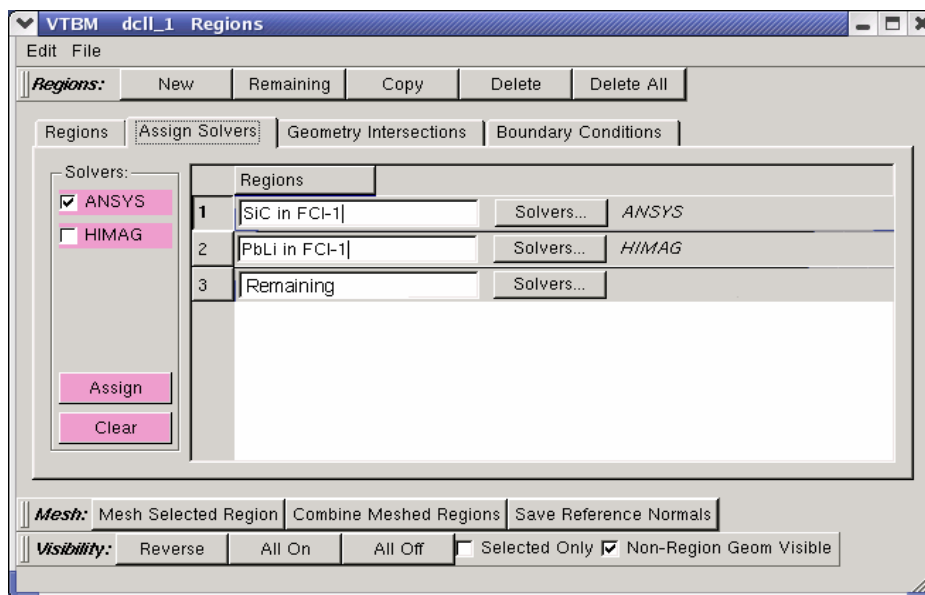


The screen can display selectively the regions that are being meshed, as below:

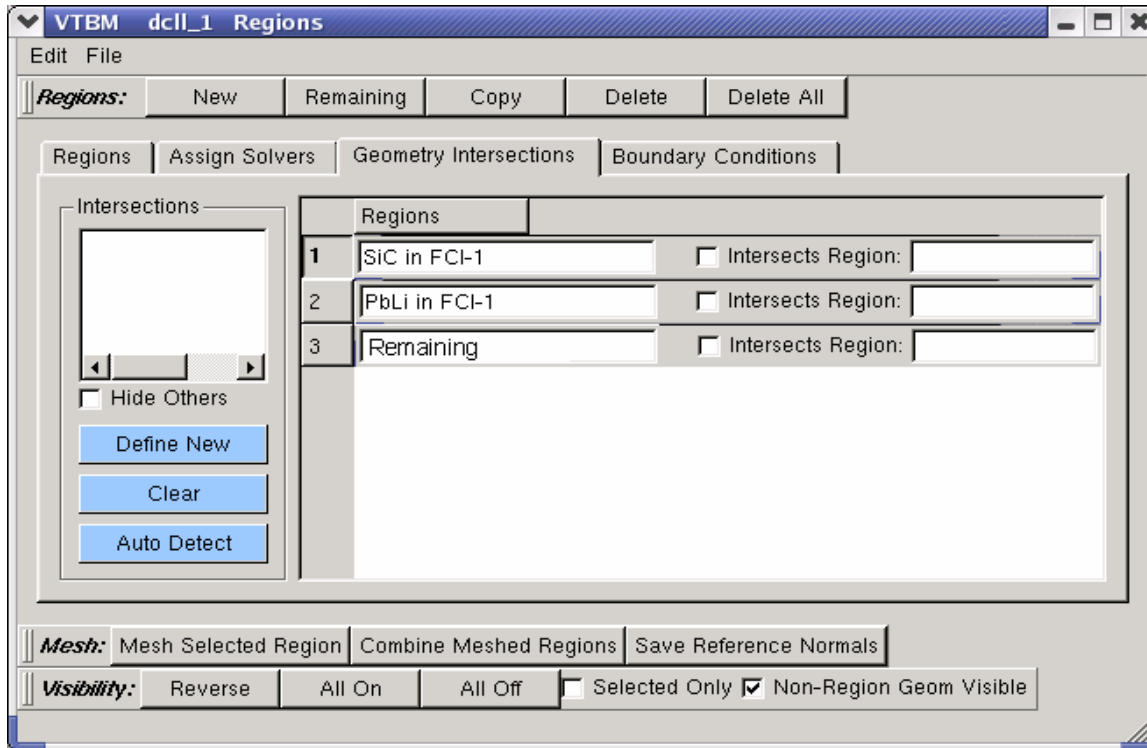


In the screen capture above the Remaining region is not visible, furthermore the “Mesh F8” checkbox is checked in the toolbar to show the default meshing for the regions.

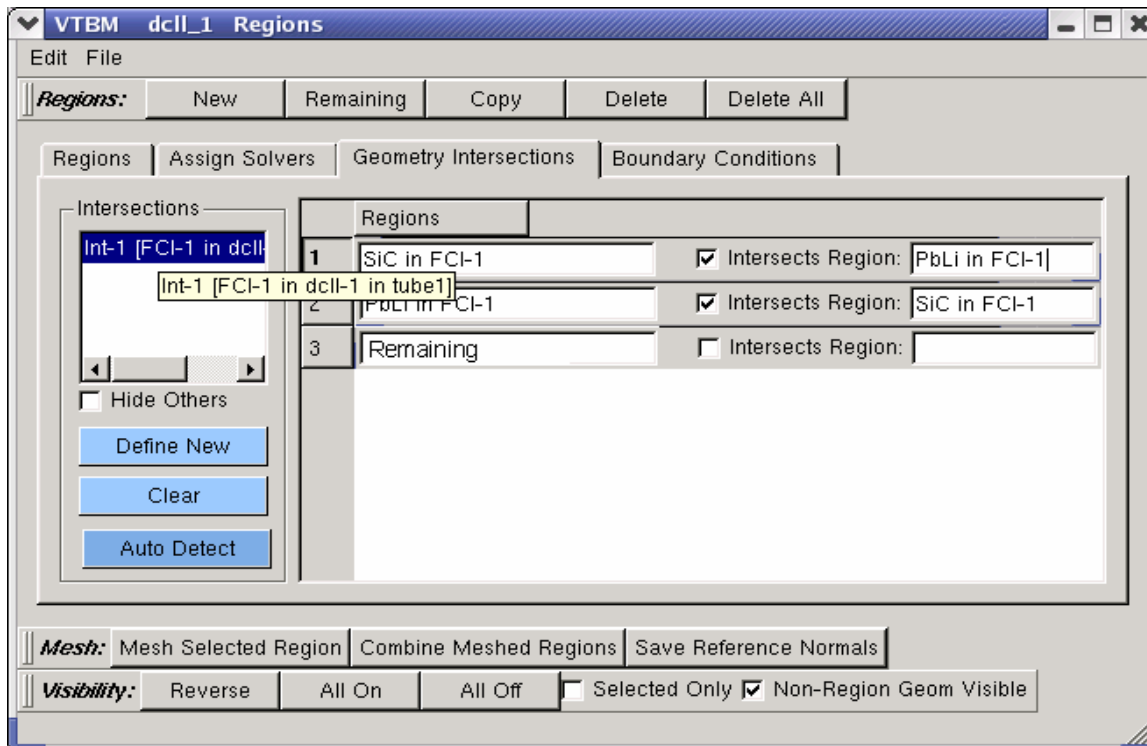
We show next the “Assign Solvers Tab of the Regions” dialog. As the region is selected from the list, the solver assigned to it is checked in the Solvers Group box. Here the SiC in FCI-1 region is assigned to ANSYS.



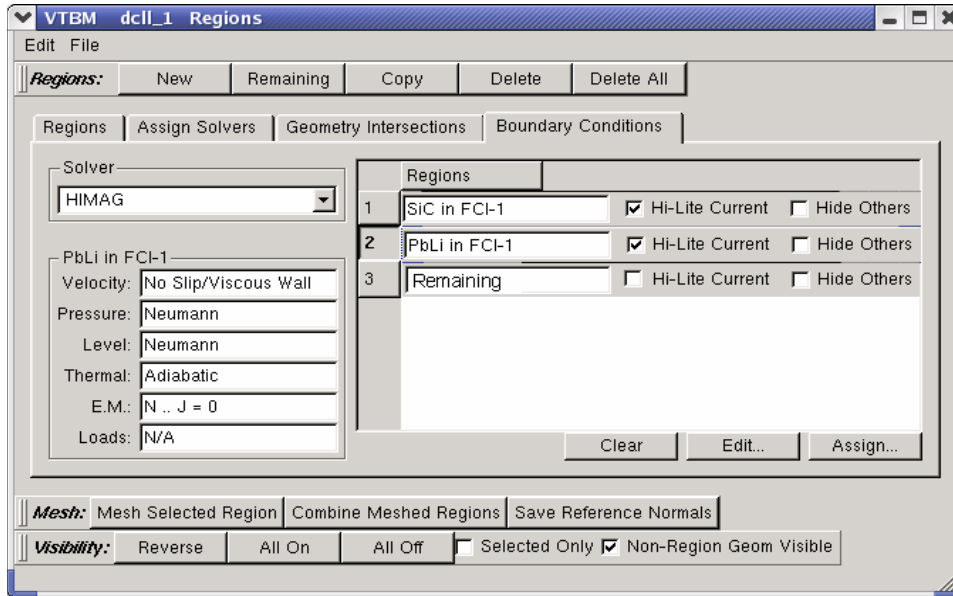
The Geometry Intersection Tab is shown below. At first the regions do not have intersections defined.



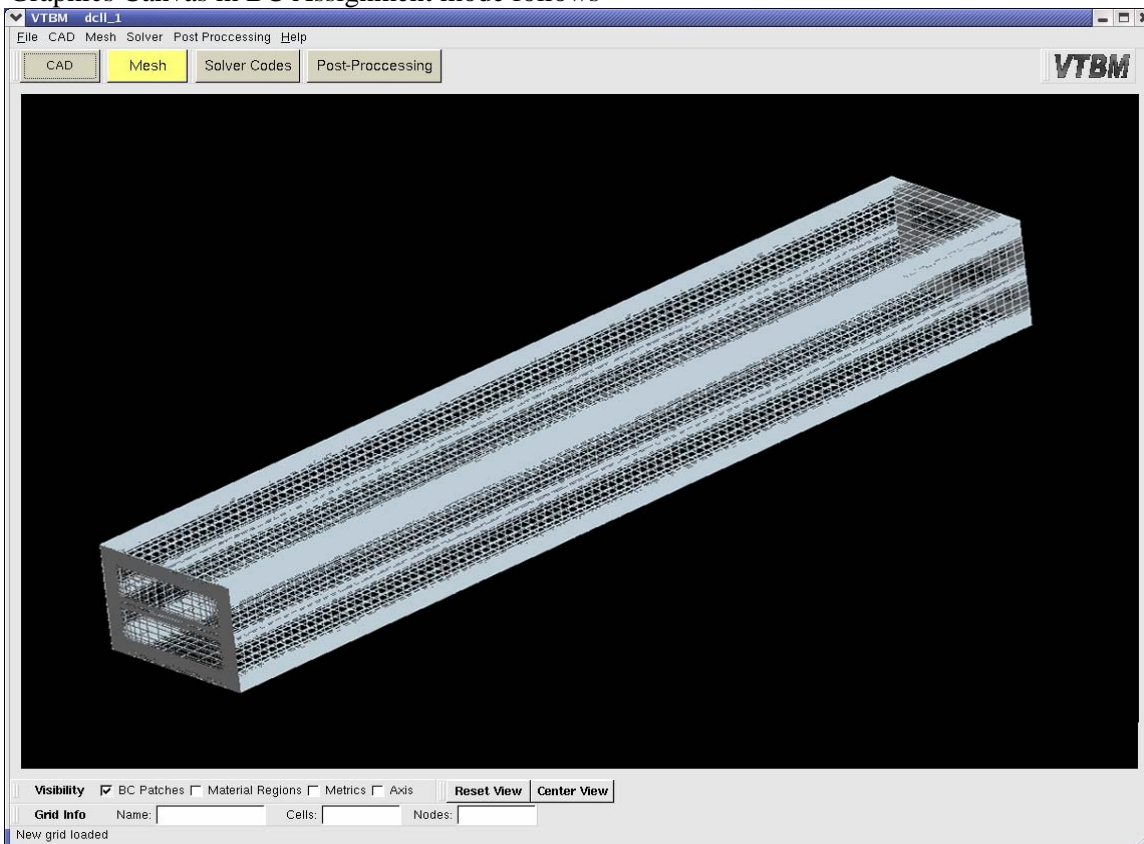
The Auto Detect function identifies intersections and automatically updates the table as seen below.

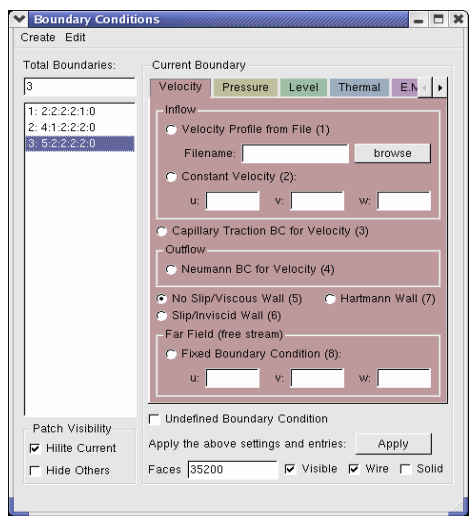


The Boundary Conditions Assignment Tab of the regions dialog. The selected region's BCs are indicated in the groupbox of the same name.

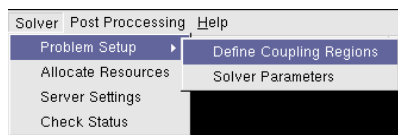


The Graphics Canvas in BC Assignment mode follows



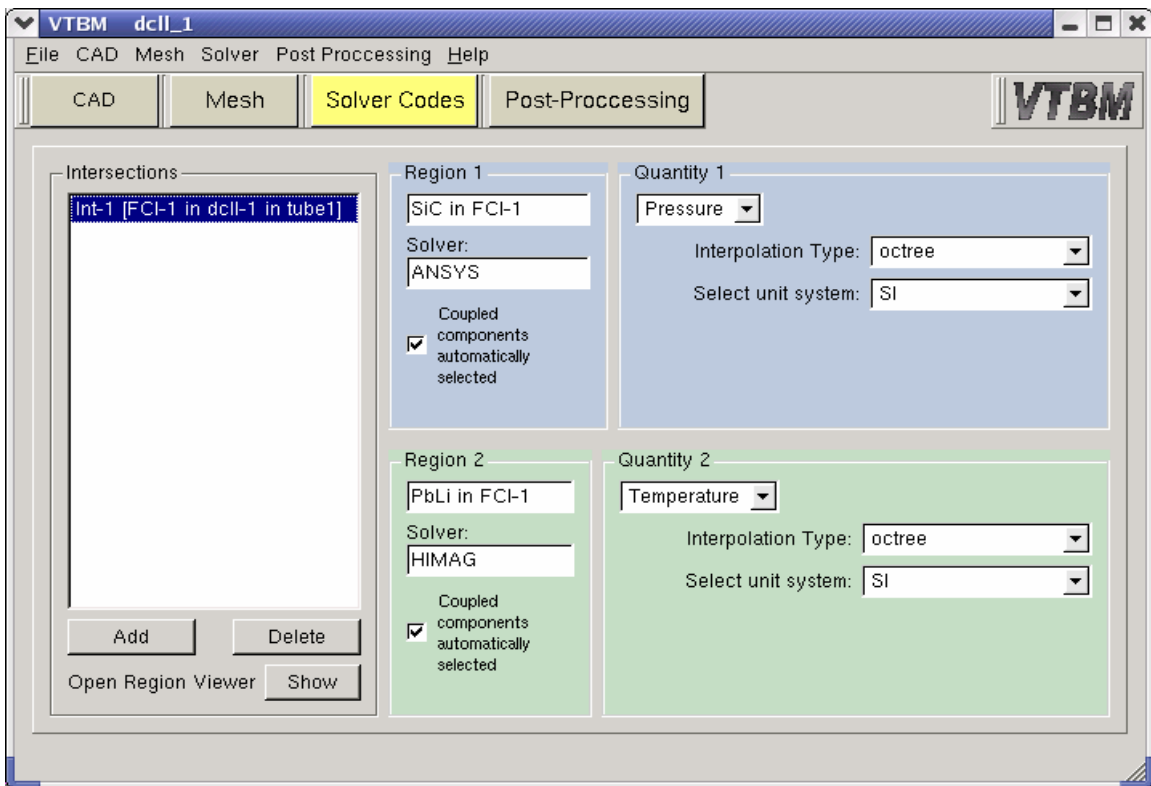


A detail from the BC dialog box is shown adjacent:



We show next how regions are coupled from the solver menu:

Coupling regions require various parameters to be specified, as below. The quantities that are to be exchanged at each coupling interface and their corresponding “to” and “from” solvers are defined.



4.3 CAD-based operations

We present here progress made during phase-I in CAD-centered functionality of the VTBM. These are broadly grouped under:

1. Management of CAD data – cleanup, editing, output
2. Data interpolation and surface deformation

An advanced graphically driven system available at HyPerComp, named TEMPUS-G, has been extended to include these features. TEMPUS-G itself is a complete surface modeling and mesh generation program for engineering simulations. During the course of this SBIR project, this software is being modified to suit VTBM needs. TEMPUS-G uses the following hierarchy for geometry representation, beginning from the definition of a curve, leading to trimmed surfaces of general shape. NURBS provides a very powerful means to described complex geometries and is widely used in surface modeling.

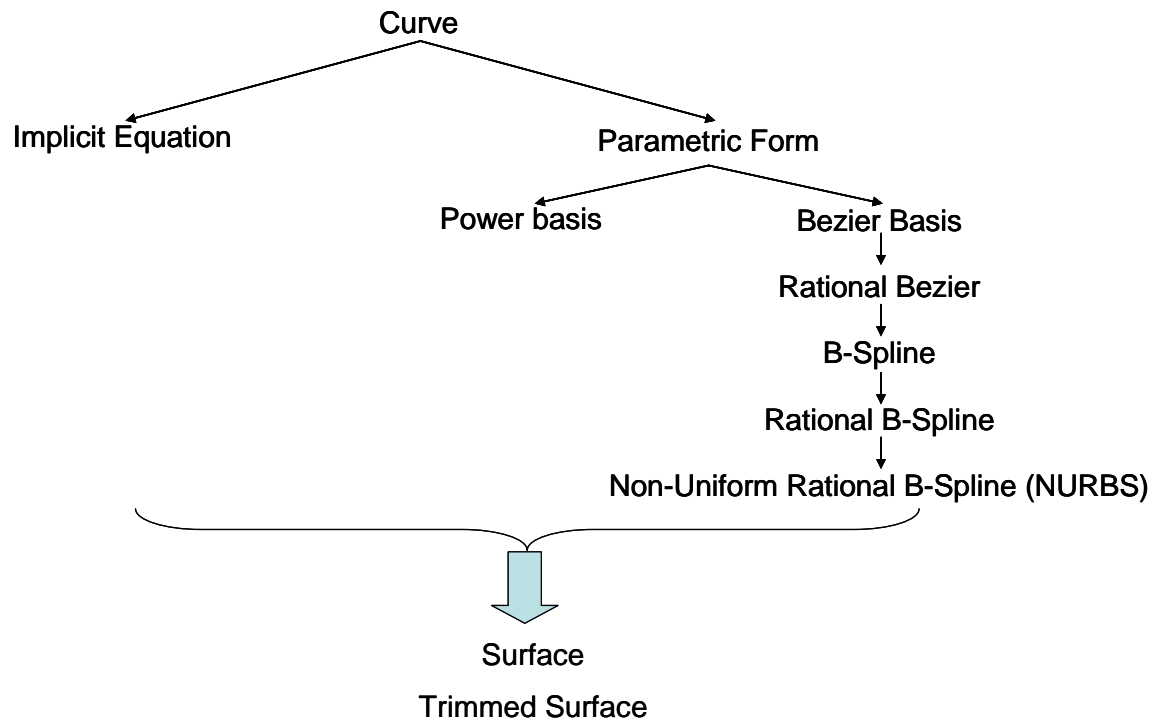


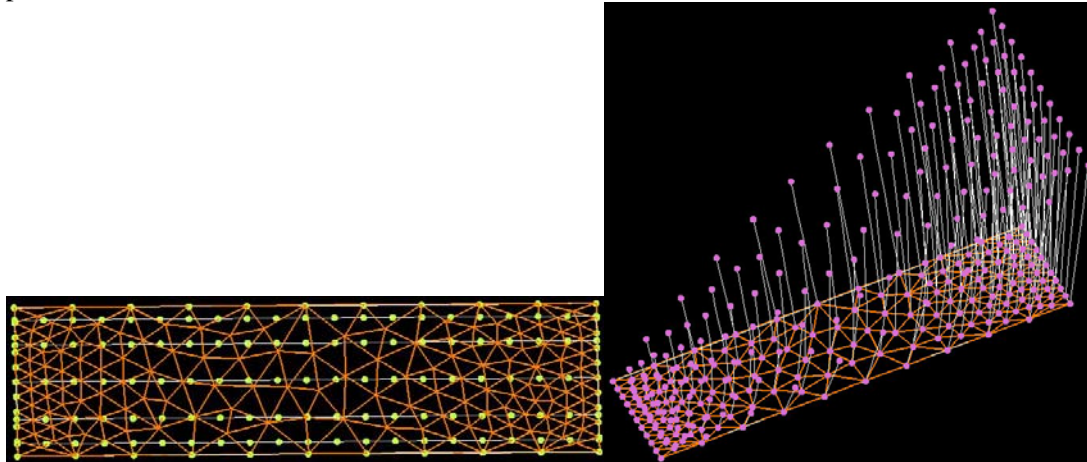
Figure 20: Hierarchy for geometry representation in TEMPUS-G/VTBM

TEMPUS-G/VTBM can presently support IGES and STEP format CAD data that can be written by all external CAD modeling software. When the CAD model (as in most instances,) is able to use non-uniform rational B-spline (NURBS) information, a general three-dimensional least squares procedure may be developed that permits good control over the tradeoff between smoothness and accuracy of the solution.

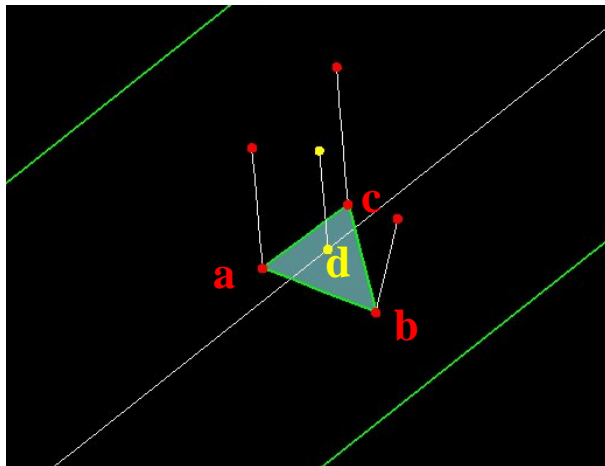
In phase-I research, we equipped the TEMPUS-G/VTBM software with the following important ability: To associate a given CAD file with two different computational meshes of widely varying resolution, by mapping the mesh points into the NURBS specification of the surface. This is accomplished by first defining a system of patches local to TEMPUS-G, and capturing the curvature and other fine details of the geometry sufficiently well. Each patch will have multiple sets of mesh points associated with it. When physical data is provided at mesh points, the NURBS functions provide a very simple and elegant method by which to interpolate this data to other points (perhaps belonging to an alternate mesh,) by serving as basis functions for interpolation. In a straightforward extension to this procedure, we find that geometric deflection provided at mesh points can be mapped into the NURBS coefficients, and the surface itself can

be deformed, thus permitting a new mesh to be generated, or the old mesh points to be relocated such that they lie exactly on the deformed surface. We find that this is a very powerful method to deal with data transfer and deformation of solid models during multiphysical modeling and describe it in further detail below, by first applying it to the deformation field.

1. We begin by obtaining the surface mesh and deflections (dx,dy,dz) at each mesh point for a given patch, as illustrated below:



2. For every point d that defines the NURBS curve, we locate the mesh cell which contains this point using an octree search, interpolate dx, dy and dz value at point d from mesh points a, b and c using 2D linear interpolation.



A: area of the element

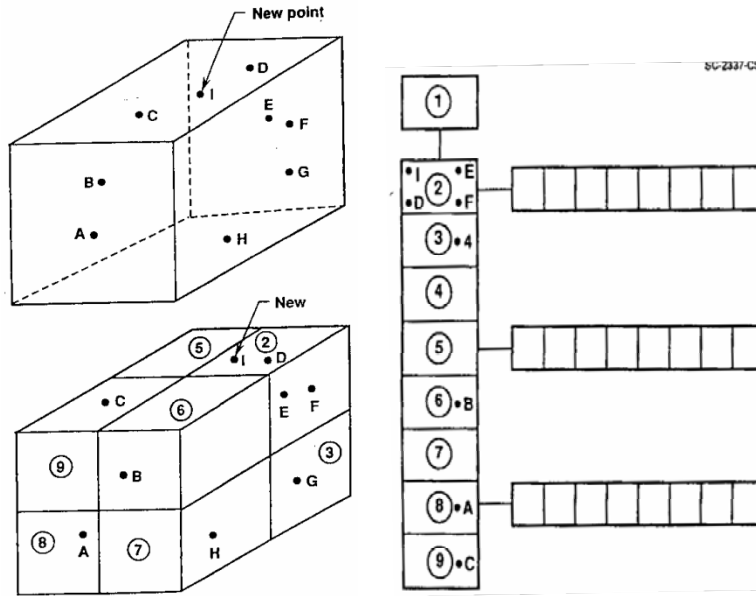
$$\phi = N_i \Phi_i + N_j \Phi_j + N_k \Phi_k$$

where:

$$N_i = \frac{1}{2A} [a_i + b_i x + c_i y] \quad \text{and} \quad \begin{cases} a_i = X_j Y_k - X_k Y_j \\ b_i = Y_j - Y_k \\ c_i = X_k - X_j \end{cases}$$

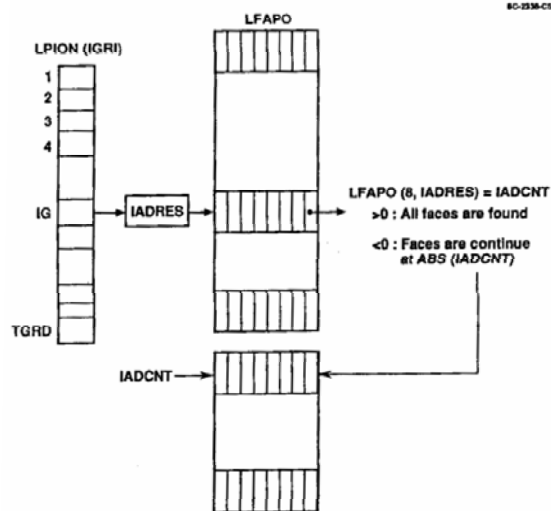
$$N_j = \frac{1}{2A} [a_j + b_j x + c_j y] \quad \text{and} \quad \begin{cases} a_j = X_k Y_i - Y_k X_i \\ b_j = Y_k - Y_i \\ c_j = X_i - X_k \end{cases}$$

Summary of the Octree search procedure: The first octant is chosen as the boundary of the geometry. At most, eight points are selected in each octant. If ninth point falls into an octant, then it is subdivided into eight smaller octants. By using this octree, the neighboring points with in a specified distance of given point can be determined very rapidly. The octree structure is shown in the following figure.



To locate the mesh element containing the geometry defining point:

- We first construct links between points and mesh elements:
- An address pointer array LPION(IPONT) for each grid point IG is first allocated.
- The faces surrounding point IG are saved in an array LFAPO(8,Iadress).
- The mesh elements surrounding a given point can quickly be found
- By using left hand rule the mesh element which contains the geometry point is then located



Deformation of the surface defining points is performed using the interpolated values of the mesh-based deformations:

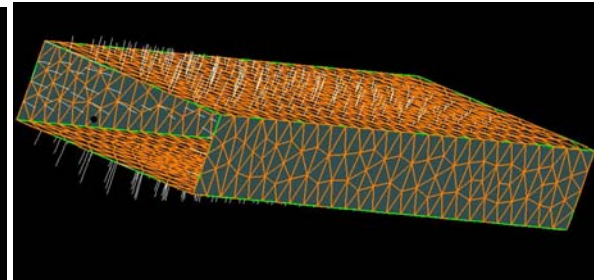
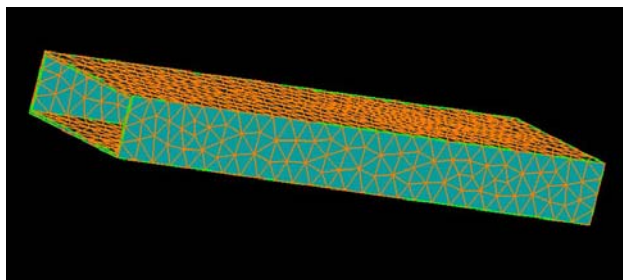
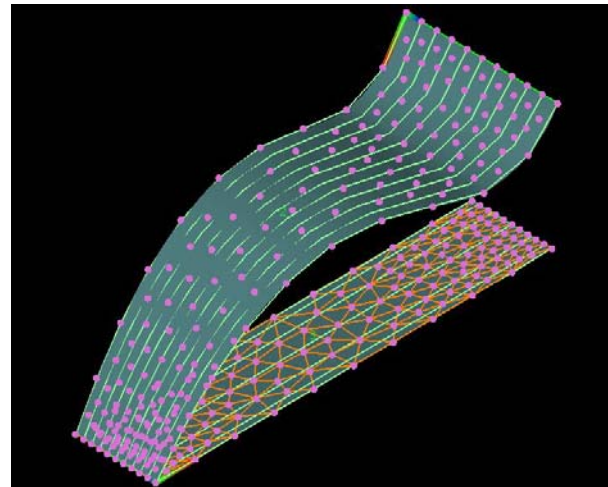
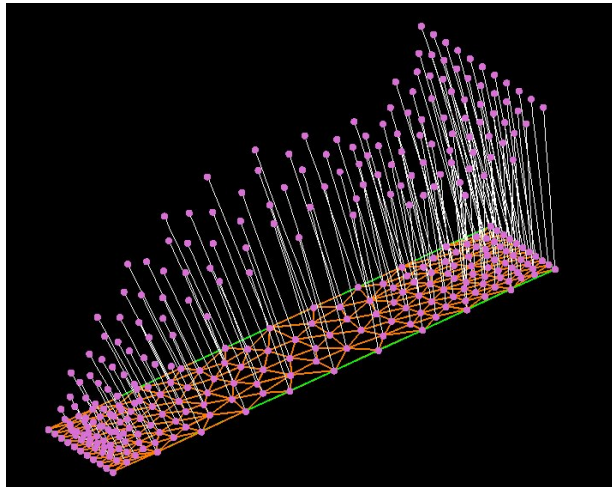
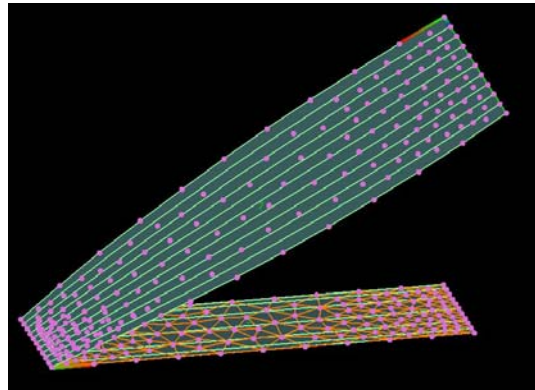
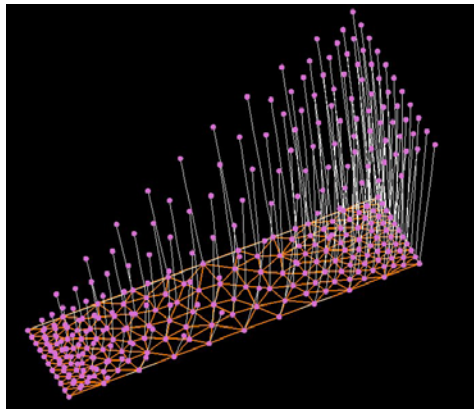
New geometry edges are generated by using

$$X(\text{new}) = X(\text{old}) + dX \text{ (interpolated)}$$

$$Y(\text{new}) = Y(\text{old}) + dY \text{ (interpolated)}$$

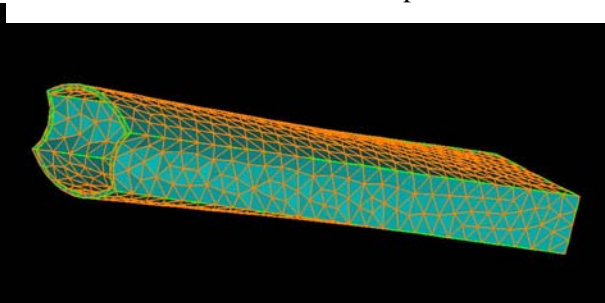
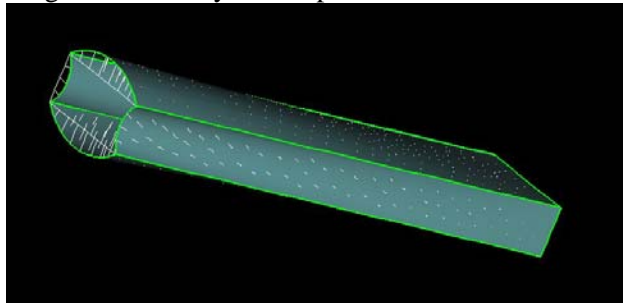
$$Z(\text{new}) = Z(\text{old}) + dZ \text{ (interpolated)}$$

New deformed surface patch is generated by these newly formed edges



Original Geometry and imported mesh

Deformation vectors at mesh points

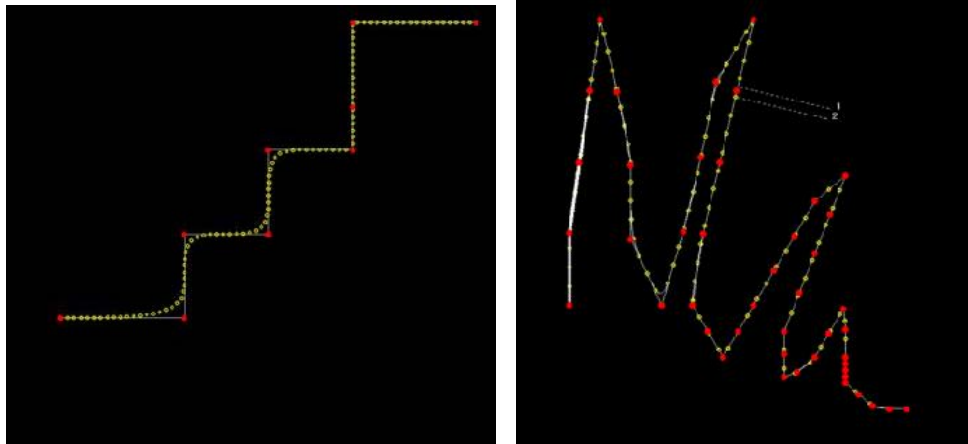
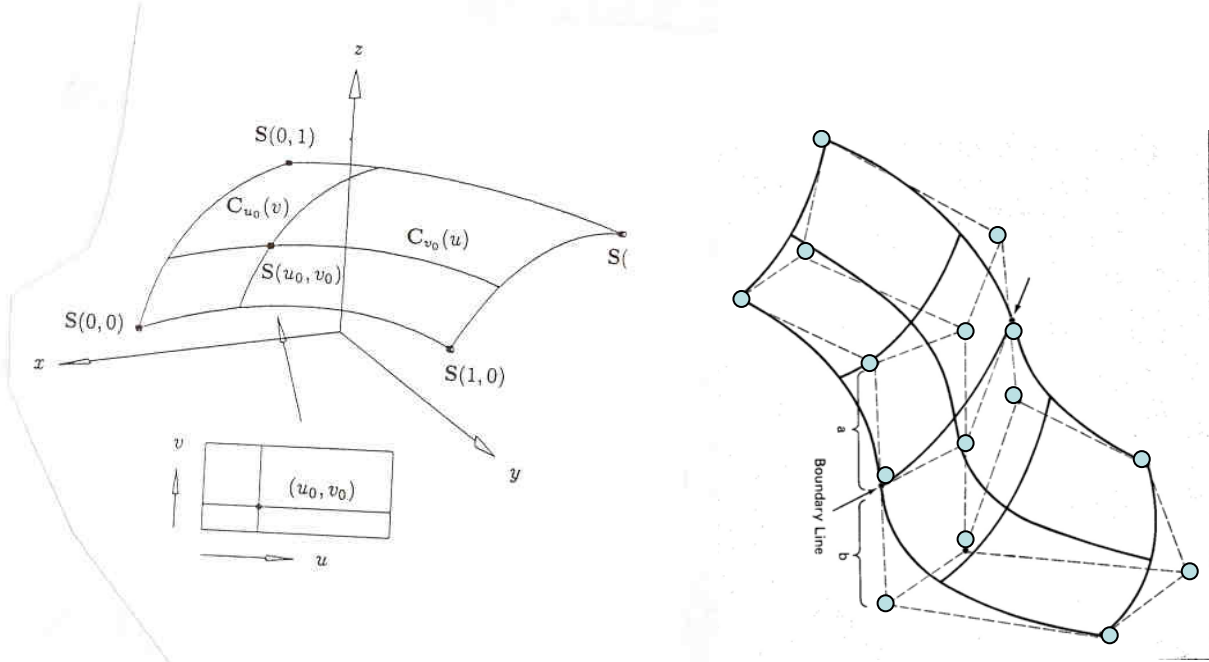


Newly generated outer box geometry

Completed deformed solid model with mesh

Figure 21: Illustrations of deformation fields and deformed patches, the last four images show a closed solid model with mesh

This development is sufficiently general and extends to very complex shapes. The interpolation functions shown above are 2-D linear basis functions. However, higher order accuracy is easily obtained, as shown below. We first note that NURBS describe a parametric variation of curves in u and v directions, which bound the NURBS patch, as shown in the figure below (left). While the control points shown on the figure on the right are used to determine the curve in an approximate sense (the surface does not have to in general pass through each of these points,) we fix a formulation wherein we can actually pass the NURBS through each of the control points, as shown contrasted in the lower figures.



The transformation metric which maps the NURBS coefficients (based on CAD) with the interior points defined by a mesh number 1 can be written as:

$$\{F_C\} = [T_{C1}]\{F_1\}$$

A similar metric is defined for a mesh number-2 based on the same NURBS representation:

$$\{F_2\} = [T_{2C}]\{F_C\}$$

Depending on the nature of the solver used (finite volume, finite element, etc.), these expressions have specialized forms, as described by Farhat et al⁵.

The interpolation procedure described earlier is then modified as follows. In the first step, the field data is mapped onto the CAD model. In the second step, a NURBS representation is used to fit the data. A NURBS surface $R(u,v)$ can be represented in a parametric form where u and v are independent parameters transcribing a surface, P_{ij} are the locations of NURBS control points, W_{ij} coefficients are the weights, u and v are the parameters, I and J are the numbers of control points in the u and v directions and $B_{ip}(u)$ and $B_{jq}(v)$ are the B-spline basis functions of degrees p and q respectively:

$$R(u,v) = \frac{\sum_i^I \sum_j^J W_{ij} B_{ip}(u) B_{jq}(v) P_{ij}}{\sum_i^I \sum_j^J W_{ij} B_{ip}(u) B_{jq}(v)}$$

The field solvers provide loads, deflection vectors and other quantities at discrete points r_m . These points are projected on the NURBS surfaces, a surface containing the point is located and the associated parameters u_m and v_m are found. Typically, this information is stored during the grid generation process itself. If the NURBS expression above is recast for the surface mesh as:

$$R(u,v) = [C]P,$$

A new NURBS surface is fitted through the modified surface data as:

$$\{F_c\} = [C^T C]^{-1} [C]^T \{F_1\}$$

The same procedure may be applied to deflections of solid boundaries in producing a new CAD model of the geometry in terms of NURBS. This development is now complete and is usable for any higher order accurate interpolation needs in TEMPUS-G/VTBM.

4.4 A Case Study in Coupled Code Execution

We present two cases using the DCLL-type FCI geometry in which we deform the geometry using ANSYS following an internal pressure distribution that is first prescribed as constant every where (case-1), and then given from calculated values from HIMAG (case-2). The procedure is as follows:

The ANSYS mesh and deformation data is obtained via a neutral (CGNS) format results file derived from the ANSYS CDB file and the output of nodal deformation data from ANSYS. The CGNS file is converted into an internal HyPerComp unstructured volume grid file (.ux format) such that we may use our existing in-house tools and utilities. A tool is used to extract the surface mesh from the volume mesh file. This is done because when we deform our CAD geometry we require only a surface mesh along with the deformation data.

At the very commencement of the simulation, we import the CAD file into our in-house CAD tool and grid generator. The CAD file is converted into geometry surfaces using an internal surface representation. Next, the surface mesh that was obtained from the CGNS data in the last step is imported into our tool. Both the geometry surfaces and surface mesh are visualized in the graphics window to ensure that they are consistent. The surface mesh should lie on top of the geometry surfaces.

Next, from within our CAD and gridding tool, we import the ANSYS deformation data from the CGNS file. A new copy of the original undeformed mesh is created with the deformations applied. The user is prompted to scale the deformation values in order to better visualize the resulting deformation. Once the user is satisfied with visualizing the deformation, the deformation is applied to the existing geometry surfaces at a push of a button. The interpolation algorithm deforms the surfaces and the results may then be examined in the graphics window. The deformed surfaces are then used to create a new volume mesh for HIMAG or other solvers.

Inter-Process Communication – A case study with HIMAG and ANSYS

1. Begin with CAD – IGES, Parasolid formats of the same geometry
2. Use HIMAG-GRID to generate blocking structure
3. Generate ux file for HIMAG and ANSYS
4. Convert ux to CDB for ANSYS
5. Setup BCs and input files for HIMAG
6. Setup CDB file with BCs and input data for ANSYS by editing template
7. Run HIMAG, output solution in CGNS, TECPLOT formats
8. Run HIMAG2CDB to extract solution along interfaces
9. Run INTERPLT to extract surface data to ANSYS mesh
10. Regenerate ANSYS CDB file with mesh, BCs and loads
11. Run ANSYS from command line to generate deflections file
12. Run DEFEX to get deflections in common data format
13. Run HIMAG-GRID to read deflections, redo CAD
14. Using existing blocking structure, regenerate mesh from new CAD
15. Write POST data if needed.
16. End of one iteration.

The above summary and the following figures illustrate this process for the cases 1 and 2 defined earlier.

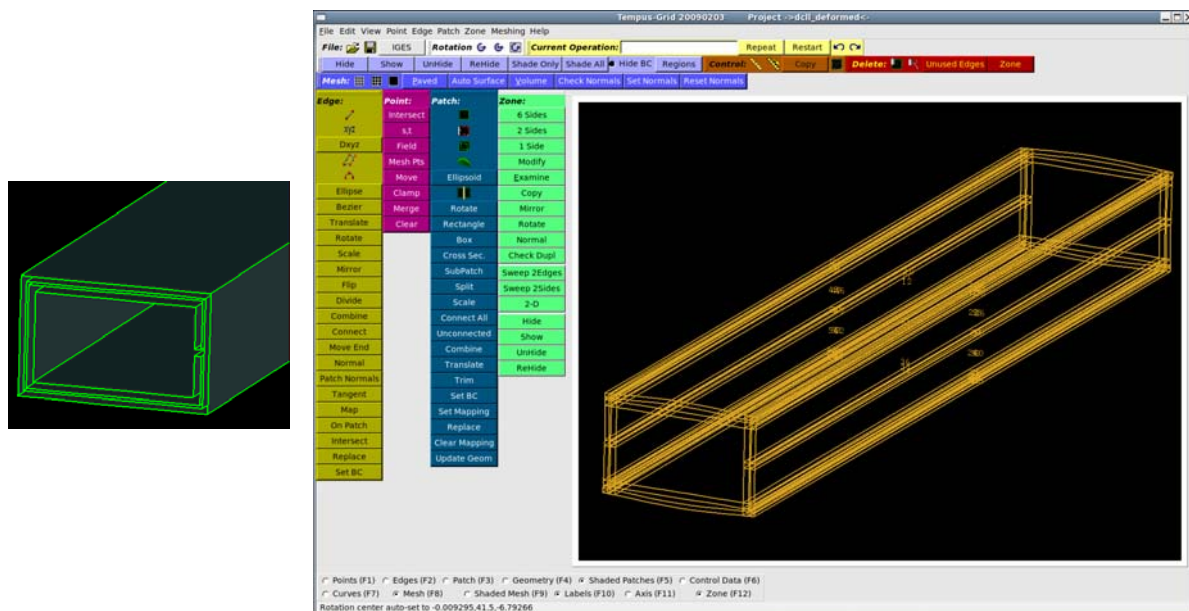


Figure 22: DCLL-FCI geometry (left) showing SiC insert within a Ferritic steel duct for PbLi flow, and (right) showing the blocking structure used for mesh generation

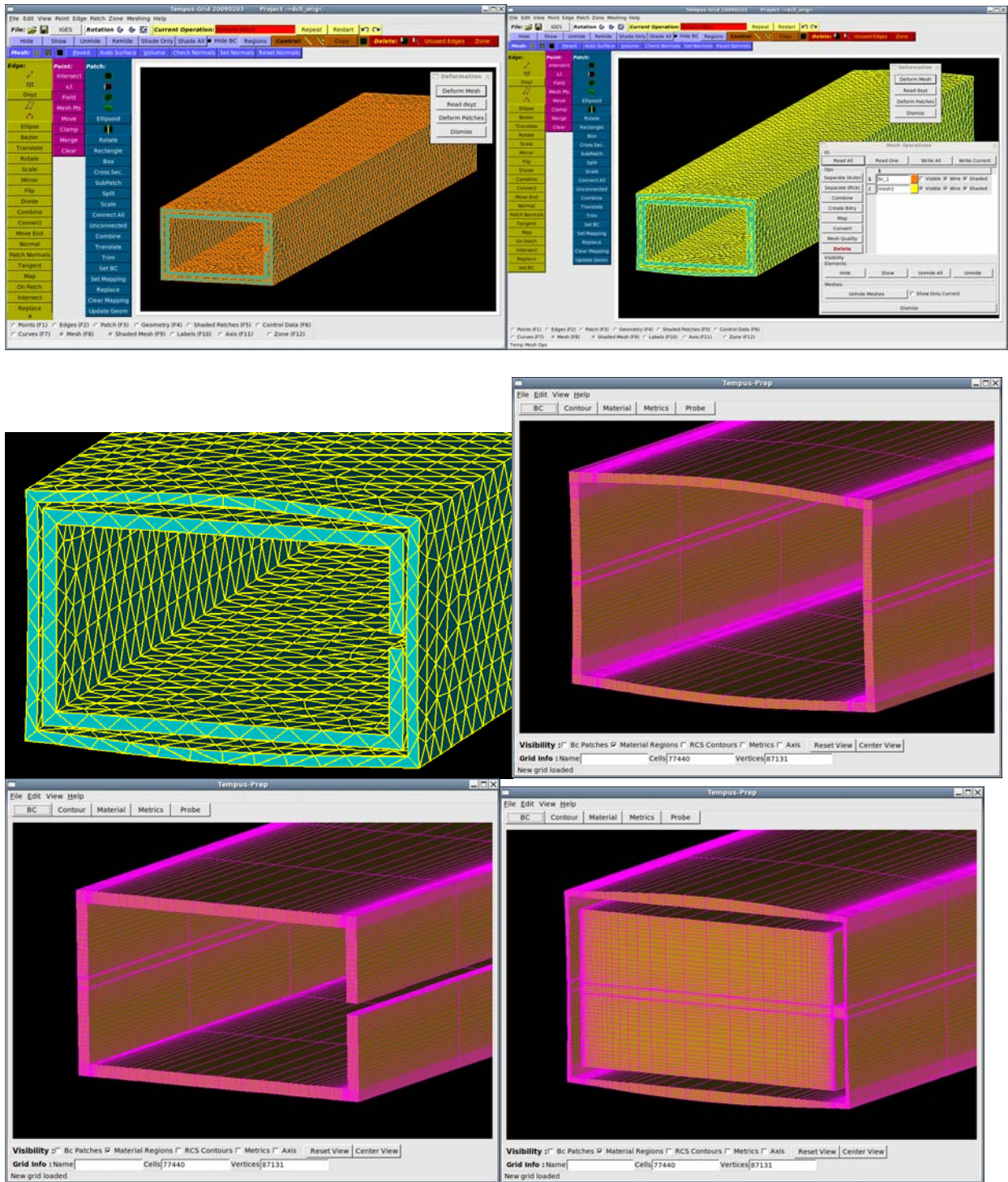


Figure 23: Case-1 (uniform pressure) from above, original mesh generated for ANSYS, showing ANSYS data extracted into a deformed surface shape automatically from TEMPUS-G, and the deformed shape being used to automatically regenerate a volume mesh for each material region.

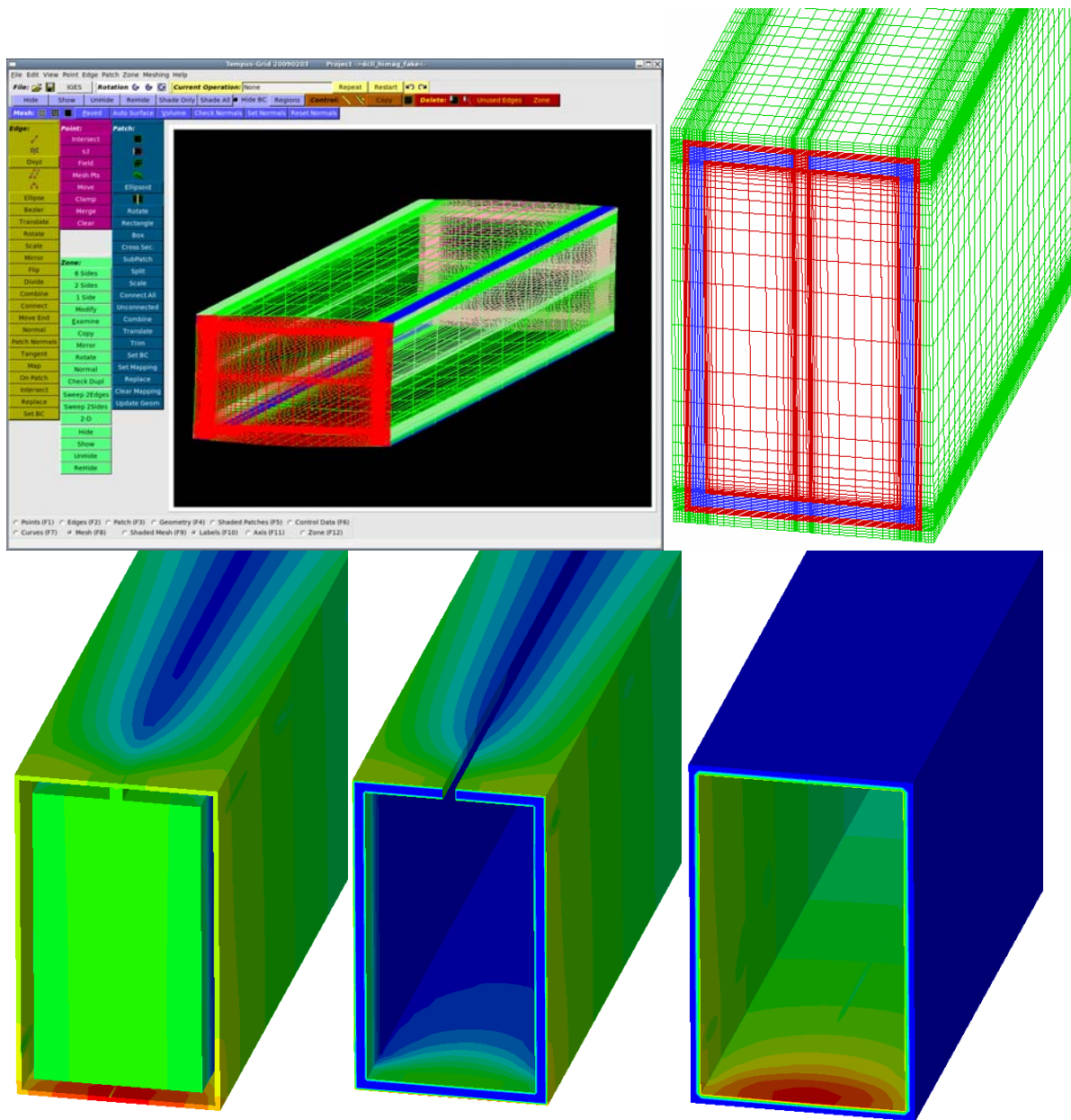


Figure 24: Case-2, involving a coupled HIMAG-ANSYS simulation, showing mesh generation, material regions and pressure contours computed using HIMAG in each material region (PbLi, SiC, FS respectively)

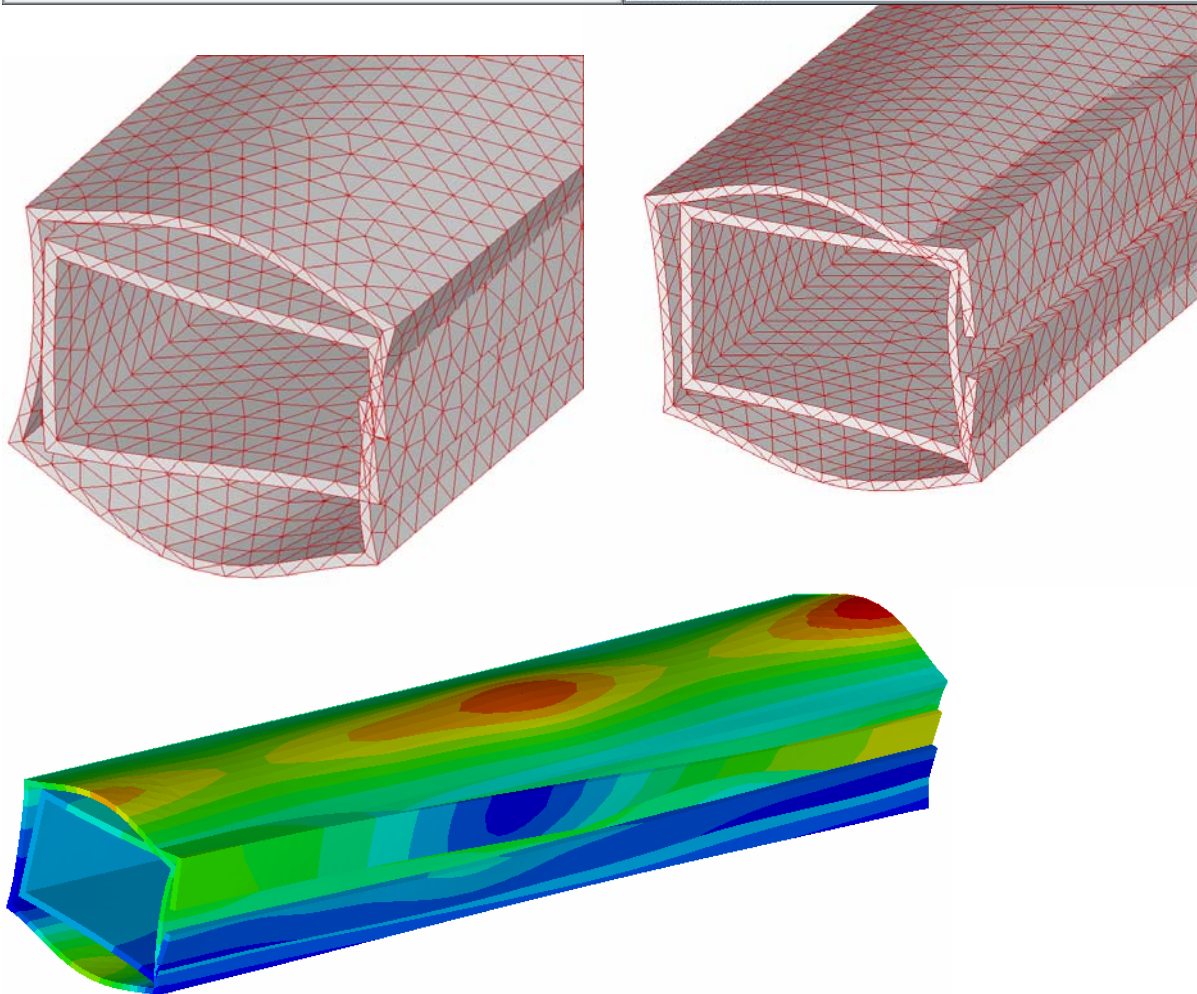
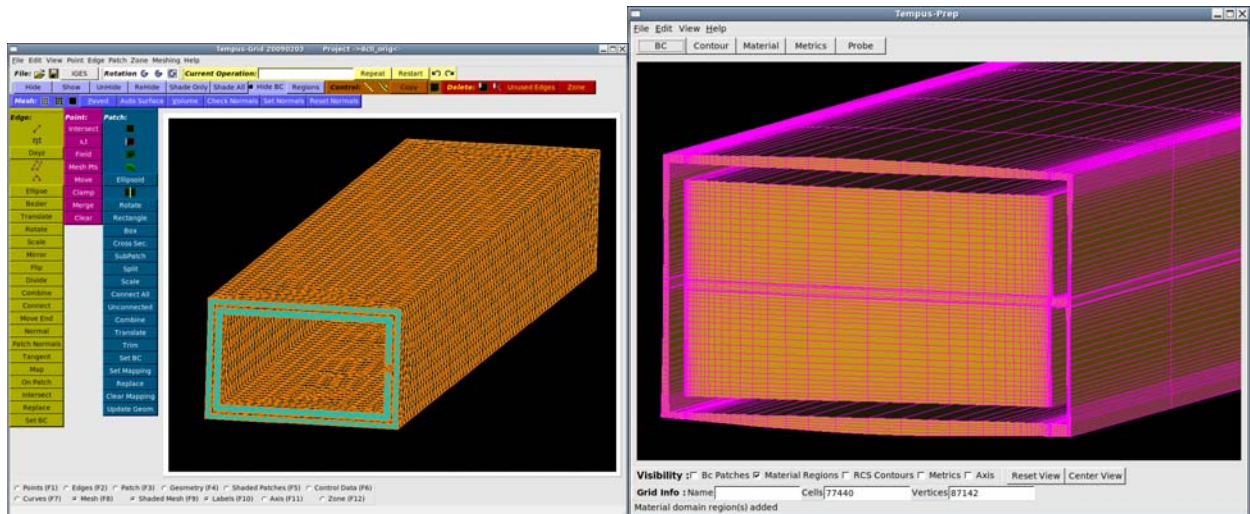


Figure 25: Original mesh and deflected mesh as computed by HIMAG/ANSYS. Note that the deflections are greatly exaggerated just to accentuate the capability and understand the mode shape for the deformation. The same utility is used to interpolate data between the two solvers.

4.5 CAD/geometry model development, template based approach

Almost all the current CAD software tools provide parametric geometry creation capability as long as a user remains in the same CAD system. This enables association among different sketches and features (extrude, revolve, cut, etc.) and helps in capturing the design intent. In order to illustrate the meaning of design intent, we present an example of missile design. Figure [26] shows a missile design created in SolidWorks with (a) original design and (b) when the underlying sketch dimension of the OML (outer mold line) is increased. A change in OML sketch dimension changes the OML radius as the OML is a surface of revolution. It can be seen from figure [26] that as the OML radius is increased, the radius of every bulkhead increases automatically to match the new surface. In the same picture, one can note that the locations and sizes of the control surfaces remain unchanged. The way some parts have changed due to the change in sketch is result of particular design intent. And, this happens because part dimensions are internally related to each other, which is also known as parametric associability. This parametric associability is lost when the data is written in non-native format.

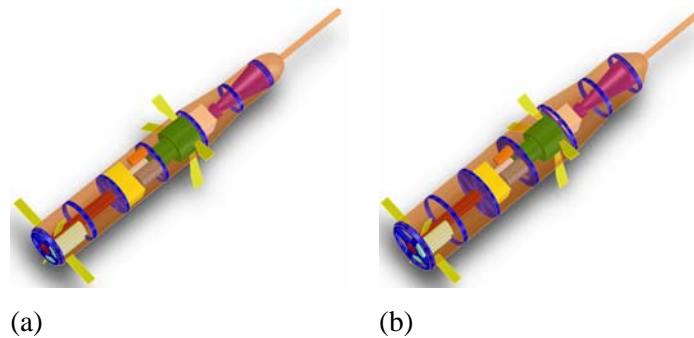


Figure 26: An example showing the effects of increased OML diameter on the missile parts.

Apart from being non-parametric, it has been found that, sometimes, a neutral CAD data file does not yield the true geometry. This is due to the differences in the tolerance levels, representation schemes, and the interpretations of the geometry among different readers and writers. For example, CSG (Constructive Solid Geometry) data structure, used in early solid modeling systems, is no longer supported by commercial systems (except, VX) today because of its inability to support complex blends and shapes. Therefore, in order to read a legacy data containing CSG data or hybrid data (combinations of B-rep and CSG), one needs to develop a specialized reader. Similarly, we find that some CAD system such as SolidWorks can read B-rep data but cannot write entities in this format. Furthermore, the accuracy and tolerance differ from one CAD system to the other. Therefore, a solid when exported from a CAD system may or may not be able to close the boundaries and create a watertight geometry due to trim inaccuracies or tolerance at the end points. Most of the CAE software requires the solids to be watertight before volumetric grid can be generated. Therefore, non-native CAD data requires further processing in order to ensure correct translation of the geometry.

An enabling component of the proposed technology is the routine usage of *template-based modeling*. The word template here refers to an intelligent parameterization of the geometry that is to be modeled, such that the user is left with broad design choices which reduce the painstaking effort in defining a composite solid object. Template-based modeling involves developing problem-specific parametrized geometries which may be readily interfaced with mesh generation and numerical modeling. Template-based modeling is a powerful, cost-saving, error-free technique, and can be customized as to include component-wise CAD modeling that is conventionally used by the designers.

While the temptation to completely parameterize the *entire* DCLL design is strong, this would limit the flexibility of the software. Alternately, we have elected to use a hybrid approach, where the template based tool is used to start off a modeling process if needed, while an existing CAD model can be imported at any time. Advanced features such as parametric associability, enforcing water-tight geometries and so forth, will be important in Phase-II research and beyond, as the software matures. It is likely also that a parametrized geometry imported from CAD systems be converted into VTBM native templates.

We developed a template based approach to HyPerComp’s mesh and CAD engine during prior investigations to study its feasibility in a VTBM. Figure [27] shows a screenshot of the graphical interface to HyPerComp’s grid generation program TEMPUS-G. Figure [10] earlier in this section discusses how the template based model may integrate with the rest of the CAD-management procedures in VTBM.

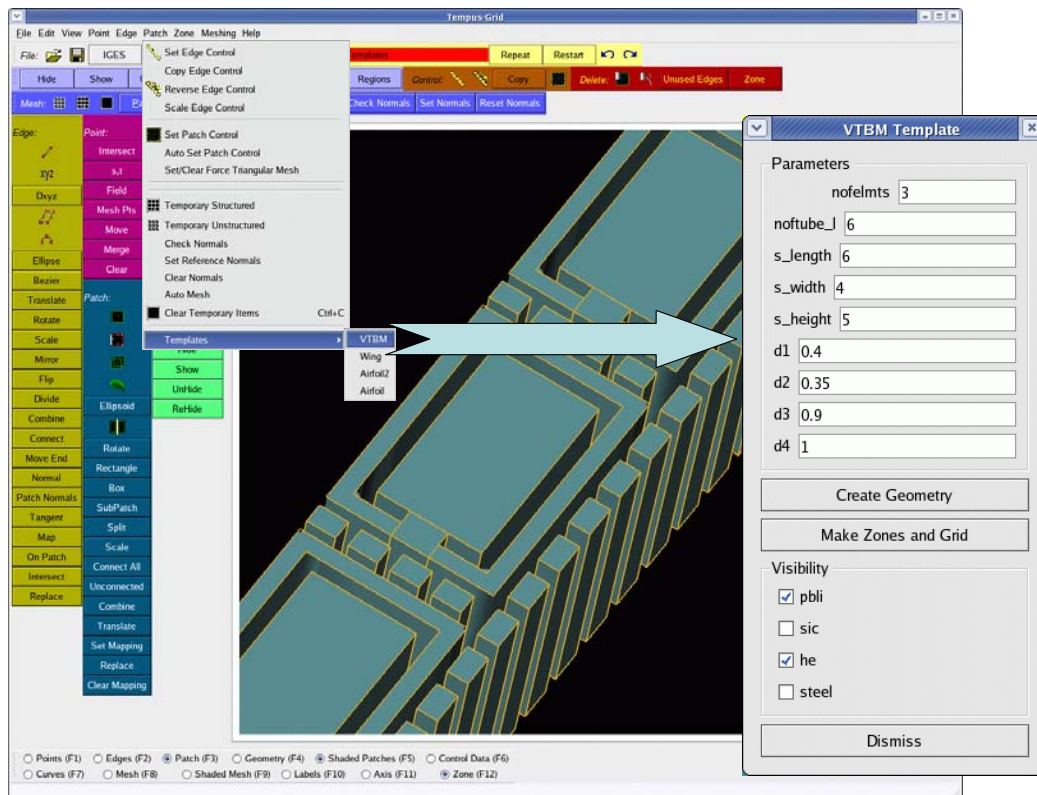


Figure 27: TEMPUS-G showing the VTBM template

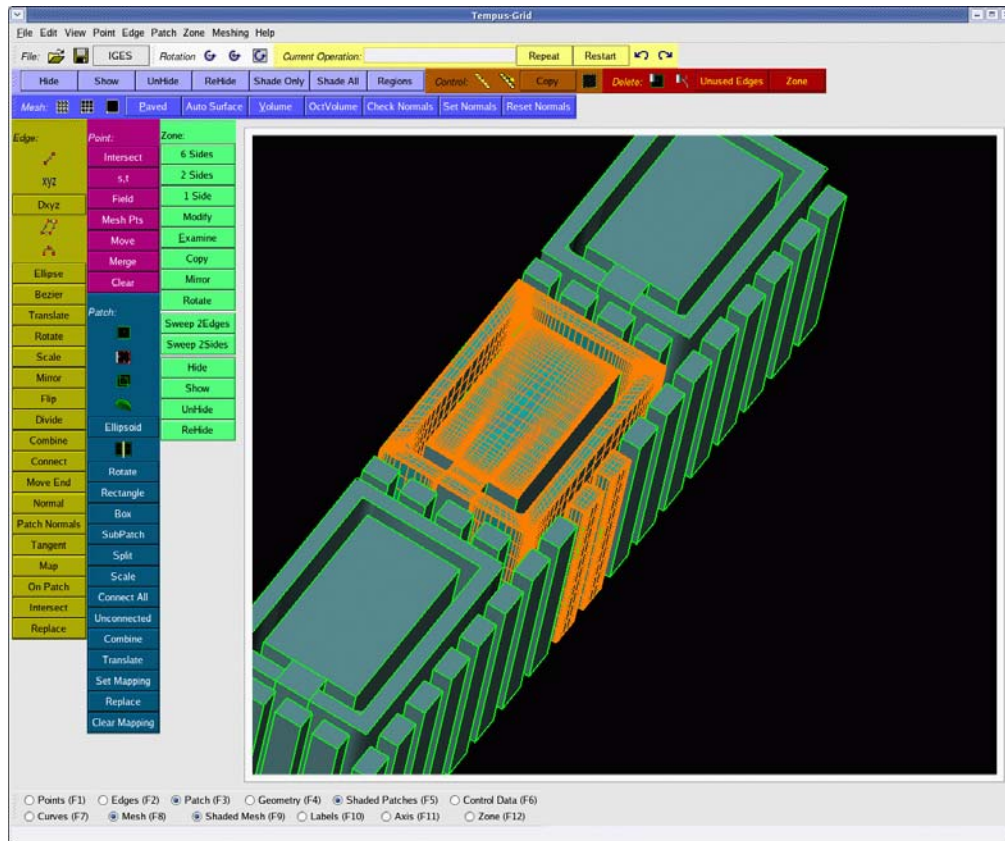


Figure 28: TEMPUS-G showing sample mesh generated

4.6 The exacting needs of mesh generation for TBM

Liquid metal MHD flow with natural convection (the principal fluid phenomenon in DCLL) is characterized by the extremely high Hartmann numbers and Grashof numbers encountered. Physically, this means that electric current and convective currents of the fluid in response to buoyant forces are concentrated in a very narrow region near the wall. In the case of MHD, the Hartmann layers which carry the greatest bulk of electric current have thickness of the order of a micron. The situation is distinguished from the fluid mechanics counterpart of boundary layers by the observation that any numerical error incurred in the Hartmann layer region is amplified by a factor of square of the Hartmann number in the core of the flow. Hartmann numbers being typically in the vicinity of 10^4 for these problems, this is a cause for immense concern. The small cells used to resolve this feature make excruciating demands on mesh resolution for full 3D problems. Furthermore, when the computational mesh is not orthogonal, corrections need to be made in the MHD solver to account for numerical errors caused by mesh skewness. Both of these factors cause a slow down of solution, and a good computational mesh can avoid these concerns to a great extent.

Figure [29] shows the direct dependence of the numerical error on mesh spacing in the Hartmann layer. We compute here the L2 norm of the error in velocity integrated over the cross section of a square duct with fully developed flow, magnetic field acting normal to the wall shown. We see that second order accuracy is preserved here in the calculation, which in itself is not guaranteed when the meshes are skewed. These observations dictate the guidelines for mesh generation in high Hartmann number problems. From our experience, MHD is the limiting constraint for TBM mesh generation, since the natural convection process is sufficiently resolved by a good MHD mesh for typical test conditions.

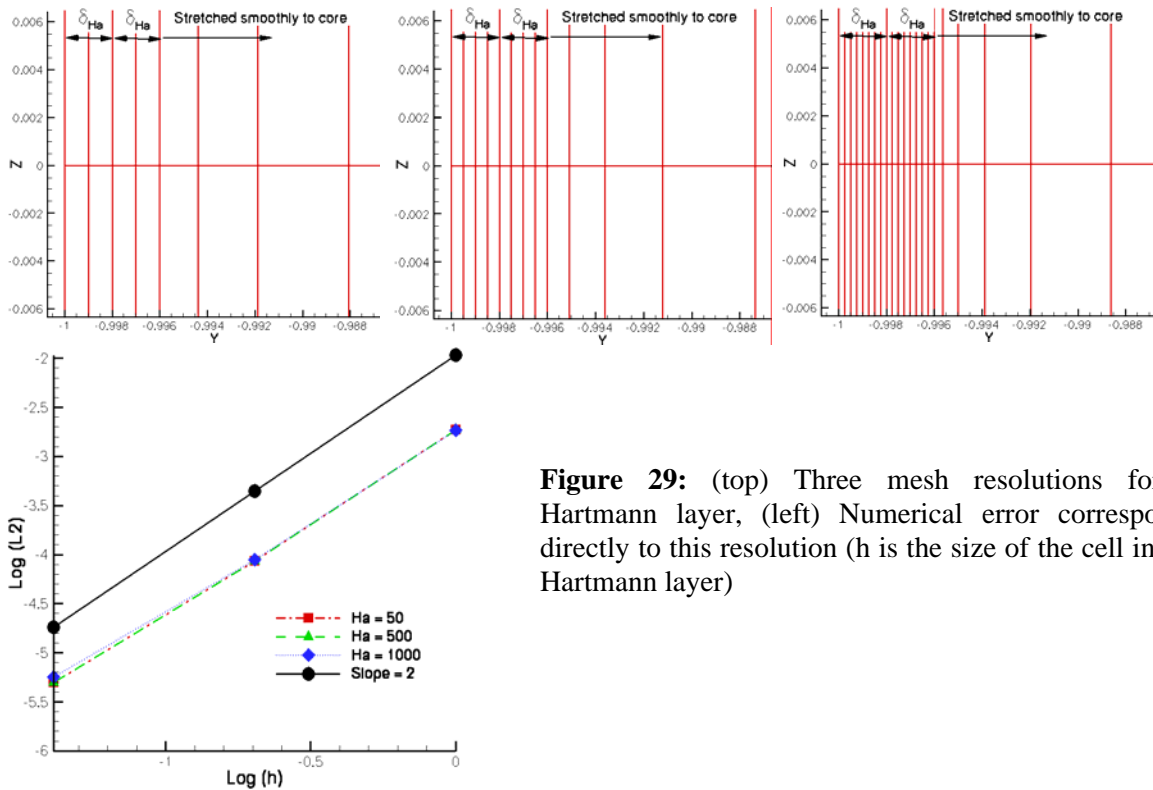


Figure 29: (top) Three mesh resolutions for a Hartmann layer, (left) Numerical error corresponds directly to this resolution (h is the size of the cell in the Hartmann layer)

Figure [30] shows the flow in a fully developed square duct with conducting walls at a Hartmann number of 10,000 computed by HIMAG, with comparisons between fully developed, 3-D and the analytical (exact) solutions to this problem. We note that the Hartmann layer thickness is extremely small, and that the side-layer shows a very large velocity peak in the near-wall region.

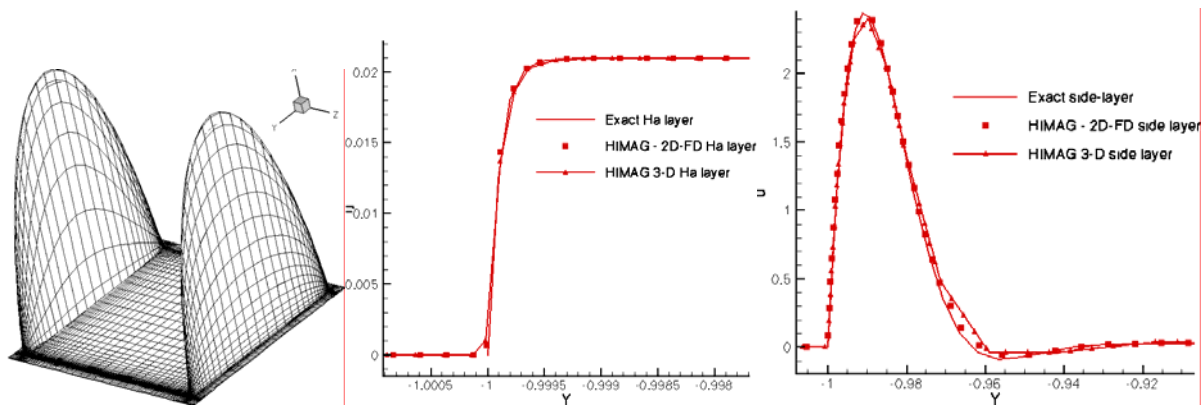


Figure 30: Velocity profile in Hartmann and side-layers at $Ha = 10,000$

A related concern exists when a high speed side-layer jet with an M-shaped velocity profile enters a narrow gap region between the SiC and Steel walls. Fig [30] shows a fully developed velocity profile in a DCLL-like cross section with three Flow Channel Inserts (FCIs). The smaller velocity peaks (which look parabolic) that are seen on the sides represent the flow in this gap region. When the aspect ratio R of this gap region is close to 1 (square cross section,) the velocity profile in this region contains strong M-shaped peaks as shown in Fig [31]. As the aspect ratio R increases and the channel becomes more slender, this

M-shape relaxes to a somewhat parabolic profile, resembling a pure hydrodynamic Poiseuille flow. These behavioral trends provide guidelines for mesh clustering in the gap regions.

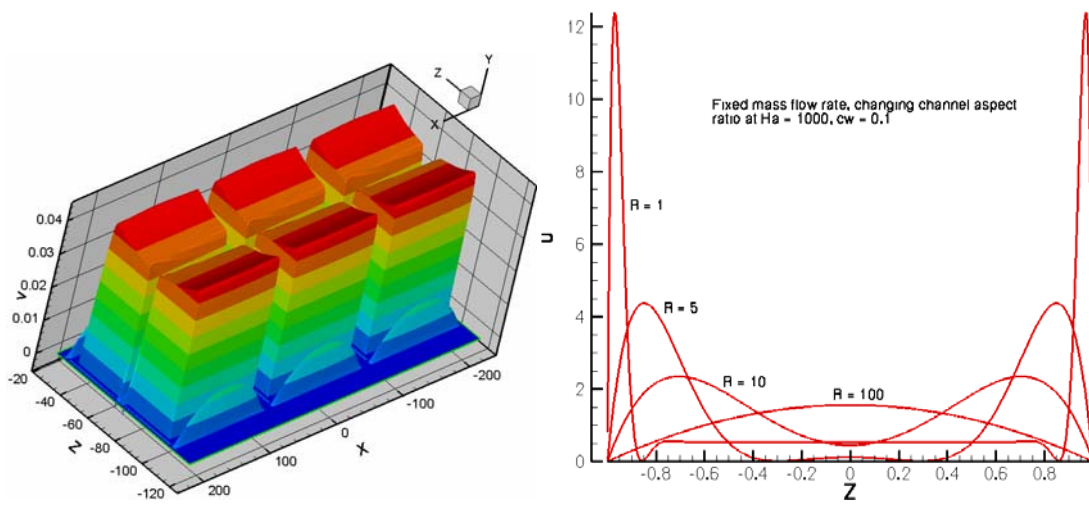


Figure 31: (left) DCLL velocity profile at $Ha = 6400$, (right) Velocity profile in the side layers as the channel aspect ratio changes, at $Ha = 1000$

4.7 A Native Format for Data Storage

In Phase-I we set out to define a native data format for the VTBM with a bias towards the use of CGNS (CFD General Notation System), which is commonly used in fluid mechanics modeling and in the aerospace community. System state descriptors and restarts will benefit from such a common format. The CGNS format is highly inclusive and flexible and is open source. Various APIs are readily available for CGNS integration. A read/write module for CGNS, translating CGNS data into HIMAG (ux and ugm), ANSYS (cdb) and SC/Tetra (PRE) formats has been completed in Phase-I. During the course of our investigation, we came across certain benefits of a more comprehensive geometry/data storage representation MOAB (Tautges [21]) which we are investigating for later use. MOAB is also an open-source format, and can deal with very complex geometric entities, apart from generalized mesh information. Further details will be discussed with phase-II plans.

A sample segment of a CGNS-type data is shown below that bears semblance to conventional CFD data structures. We present a few aspects of the CGNS data structure and components in this section.

```

file_open( "project.tbm" )

set_number_of_zones( N )

loop over each zone:
{
    List Nodal Coordinates
    List connectivity
}

set_current_zone (n)

// zonal data
// for each zone where data is available

```

```

cell_data type < int, float, double, string >
node_data type
edge_data type

type
number_of_values
values

// global data

same format as zonal data

// boundary conditions (optional)

loop over patches:
{
    boundary_type
    boundary_group_type
    boundary_sub_group
    number_of_values
    values
}

// volume conditions (optional)

loop over zone types:
{
    field descriptor
}

//end of file

```

CFD General Notation System (CGNS)

CGNS is a well-established, stable format with world-wide acceptance, use and support. It provides seamless communication of data between applications, sites, and system architectures. It is supported by most commercial visualization and CFD vendors. It is extensible and flexible and is easily adapted to other fields of computational physics through specification in the SIDS. It is backwards compatible with previous versions – forwards compatible within the major release number. It allows new software development to focus on functionality and reliability rather than data I/O, storage and compatibility

The objectives of CGNS are:

- Provide a general, portable and extensible standard for the storing and retrieval of CFD analysis data
- Offer seamless communication of CFD analysis data between sites, applications and system architectures
- Eliminate the overhead costs due to file translation and multiplicity of data sets in various formats
- Provide free, open software – GNU Lesser General Public License

Principal target is the data normally associated with fluid flows, but is applicable to computational field physics in general with augmentation of the data definitions and storage conventions.

CGNS components:

The following components are distributed with CGNS and can be readily used in VTBM development:
Advanced Data Format (ADF)

Software that performs the I/O operations
 Directed graph based on a single data structure (the ADF node)
 Defines how data is organized in the storage media.

Standard Interface Data Structures (SIDS)
 Collection of conventions and definitions that defines the intellectual content of CFD-related data.
 Independent of the physical file format

SIDS to ADF Mapping
 Defines how the SIDS is represented in ADF

CGNS Mid-Level Library (MLL)
 High level Application Programming Interface (API) which conforms closely to the SIDS
 Built on top of ADF and does not perform any direct I/O operation

CGNS Main features

The following are the main technical aspects of the CGNS format which make it an attractive candidate for the native data storage in VTBM:

- Hierarchical data structure: quickly traversed and sorted, no need to process irrelevant data
- Complete and explicit problem description
- Standardized naming conventions
- Unlimited internal documentation, and application specific data
- Layered so that much of the data structures are optional
- ADF database: universal and self describing
- Based on a single data structure called an ADF node
- The data may encompass several files through the use of links
- Portable ANSI C software, with complete Fortran and C interfaces
- Files stored in compact C binary format
- Complete and architecture independent API

4.8 Format Conversion

In order to emphasize the role of format conversion and data translation, we note that the software SC/Tetra uses the following files in its operation importantly, besides other files related to data parallelization, and specialized output:

PREI	Input to the mesh generation program
PREO	Mesh related restart file
VFI	Input file related to radiation computations
VFO	Output files for radiation computations
FLD	Output files with results of calculations
AVS	AVS-formatted post-processing file
FVW	FieldView output file
ENS	Ensign output file
TM	Time history data
RI	Input file for restart calculations
RO	Output files for restart calculations
INI	Input files for initial calculations
CUR	Output files with time history of acoustic pressure
PFO	Time history of pressure on selected surfaces

Clearly, the management of such data across multiple codes and to communicate between one another can be a formidable task. There are two principal ways to achieve this. The first is to use a “native” intermediary data format through which data is pipelined. This will be explored in Phase-II, and is generally an important aspect of standardizing multi-disciplinary simulation data. The second approach, which is nevertheless an important building block of the whole process, is to be able to read various formats and output data in other formats. This is an indispensable feature of the VTBM and much effort has been expended in Phase-I to be able to handle and exchange data in the following formats:

TEMPUS-G

IGES, STEP CAD files, and native TGP format

HIMAG

UX	Unstructured Grid
UGM	Boundary patches and BC info
MAT.BIN	Material region information
INPUT	Numerical and Physical problem specification
COLOR	Parallel partitioning information
PATCH.LIST	Solution surfaces on which interpolated data is desired or controls are computed

SC/Tetra

PRE	Computational Mesh including material regions
FLD	Field data, including computational mesh and other physical information

ANSYS

CDB	Common database format including mesh and field solution
-----	--

Utilities for inter-conversion are built into the GUI.

5.0 Conclusion and Future Plans

The development described in section [4] represents a significant first step in designing a practical VTBM environment as conceived in the SBIR objectives. The CAD-based interpolation and data transfer facility is now quite mature, and data translation protocols across the various solvers, including a native CGNS format within VTBM are now ready to be extended into an automated production level management software during any future studies. In the next section, we present the tasks and a timeline for maturing this technology into a pre-release stage for initial applications by fusion researchers.

At the conclusion of the phase-I research we have various enabling components of a VTBM environment which can be integrated into a fully functional software with graphically integrated interoperating multiphysical solvers. A phase-II proposal was submitted with the blueprint of the technology as it is perceived in various stages of maturation. We believe that fusion as well as other aspects of nuclear engineering, including complex areas of aerospace engineering design will benefit from such a utility when it is finally available.

6.0 References

- [1] Abdou M.A., Sze, D., Wong, C., Sawan, M., Ying, A., Morley, N.B., Malang, S., "US plans and strategy for ITER blanket testing," *Fusion Science and Technology*, Vol. 47, No. 3, pg. 475-487 (2005)
- [2] ANSYS - Multiphysics Package, <http://www.ansys.com/products/multiphysics.asp>
- [3] Caracappa, P.F., Xu, X.G., "Methods of integrating computer aided design for Monte Carlo simulations," *The Monte Carlo method: Versatility unbounded in a dynamic computing world*, ANS conference, April 2005
- [4] Dagher, M., Marriott, E., Wong, C., Malang, S., Sawan, M., "DCLL TBM Design Status," presentation at FNST meeting, UCLA, August 12-14, 2008
- [5] Farhat, C., Lesoinne, M., LeTallec, P., "Load and motion transfer algorithms for fluid/structure interaction problems with non-matching discrete interfaces: Momentum and energy conservation, optimal discretization and application to aeroelasticity," *Comput. Methods Appl. Mech. Engrg.*, Vol. 157, pp. 95-114 (1998)
- [6] Fischer, U., Chen, Y., Loughlin, M., Perel, R.L., Pereslavtsev, P., Petrizzi, L., Simakov, S.P., Tautges, T.J., Tsige-Tamirat, H., Wilson, P.P.H., "Advanced computational tools and methods for nuclear analyses of fusion technology systems," 20th IAEA Fusion Energy Conference, Portugal, November 2004
- [7] ITAPS – MOAB project Wiki: <http://trac.mcs.anl.gov/projects/ITAPS/wiki/MOAB>
- [8] Jambunathan, K., Lai, E., Hartle, S.L., Button, B.L., "Development of an intelligent front-end for a computational fluid dynamics package," *Artificial Intelligence in Engineering*, V. 6., No. 1, pg. 27-35, 1991
- [9] Kingsley, G., Siegel, J.M., Harrand, V.J., Lawrence, C., Luker, J.L., "Development of a multi-disciplinary computing environment (MDICE)," AIAA Paper 98-4738, 1998
- [10] Lytle, J.K., "The numerical propulsion system simulation: an overview," NASA TM-2000-209915, 2000
- [11] Maman, N., Farhat, C., "Matching fluid and structure meshes for aeroelastic computations: A parallel approach," *Computers & Structures*, V. 54, No. 4, pp. 779-785, 1995
- [12] McConnell, S., "Code Complete", Microsoft Press, Second Edition, 2004
- [13] MCNP web page: <http://mcnp-green.lanl.gov/index.html>
- [14] Morley, N.B., Smolentsev, S., Munipalli, R., Ni, M., Gao, D., Abdou, M.A., "Modeling for liquid metal free surface MHD flow for fusion liquid walls," *Fusion Engineering and Design*, Vol. 72, pp. 3-34, 2004
- [15] Narula M., Ying, A., Hunt, R., Abdou, M.A., "Integrated thermo fluid-thermal stress analysis approach for an effective design of ITER-TBM (Test Blanket Module)," SOFT, 2008
- [16] Ni, M.-J., Munipalli R., Morley, N.B., Abdou, M.A., "Validation strategies for free surface MHD flows for fusion applications," *Fusion Engineering and Design*, V. 81, pg. 1535-1541, 2005
- [17] Reed, C.B., Picologlou, B.F., Hua, T.Q., Walker, J.S., "ALEX results – A comparison of measurements from a round and a rectangular duct with 3-D code predictions," IEEE 12th Symposium on Fusion Engineering, Monterey, CA, October 1987
- [18] Samareh, J.A., "Usage of CAD geometry in MDO," AIAA paper 96-3991 (1996)
- [19] Samareh, J.A., Bhatia, K.G., "A unified approach to modeling multidisciplinary interactions," AIAA paper 2000-4704 (2000)
- [20] Sawan, M., Smith, B., Marriott, E., Wilson, P., "Three-dimensional Nuclear Analysis for the US DCLL TBM," memo dated February 24, 2009

- [21] Tautges, T.J., Kraftcheck, J.A., “Geometry Short Course,” 12th International Meshing Roundtable, Sept. 14, 2003
- [22] Tillack, M, Wang, X.R., Pulsifer, J., Malang, S., Sze D.K., Billone, M., Sviatoslavsky, I., ARIES Team, “Fusion power code engineering for the ARIES-ST power plant,” Fusion Engineering and Design, Vol. 65, pp. 215-261 (2003)
- [23] Townsend, J.C., Salas, A.O., “Managing MDO software development projects,” AIAA paper 2002-5442, 2002
- [24] Wesley, L.P., Lee, J.D., Rodman, L.C., Childs, R.E., “Toward an integrated CFD expert system environment,” AIAA paper 98-1005, 1998
- [25] Wesley, L.P., Childs, R.E., LaBozzetta, W.F., Reyhner, T.A., “Dynamic control of the WIND CFD code,” IEEE Aerospace Conference Proceedings, Vol. 6, pg. 2775-2791, 2001
- [26] Ying, A., Abdou, M., Smolentsev, S., Munipalli R., Youchison, D., Wilson, P., Sawan, M., Merrill, B., “Integrated multiphysics simulation of nuclear components as an essential element in developing predictive capabilities for DEMO,” presentation made at ReNeW workshop at UCLA, March 2009
- [27] Ying, A., M. Abdou, C. Wong, S. Malang, N. Morley, M. Sawan, B. Merrill, D. Sze, R. Kurtz, S. Willms, M. Ulrickson, S. Zinkle, "An Overview of US ITER Test Blanket Module Program", Fusion Engineering & Design, 81:(1-7), 433-441, 2006
- [28] Fusion Simulation Project, see, e.g., www.isofs.info/FSP_Final_Report.pdf
- [29] Sharafat, S., et al, “A damage-mechanics based approach to structural design of ITER components, ” US-ITER TBM meeting, UCLA, Nov. 3-5, 2003
- [30] Ni, M.-J., Munipalli R., Huang, P.-Y., Morley, N.B., Abdou, M.A., “A current density conservative scheme for incompressible MD flows at a low magnetic Reynolds number (in two parts,)” J. Comp. Phys., Vol. 227, pp. 174-228, 2007